

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ÇİFT KAYNAK KISITLI HÜCRESEL ÜRETİM SİSTEMİNDE PARTİ VE
ÇALIŞAN ÇİZELGELEME PROBLEMLERİ İÇİN BÜTÜNSEL BİR
YÖNTEMBİLİM VE UYGULAMASI**

DOKTORA TEZİ

Ömer Faruk YILMAZ

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

KASIM 2017

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ÇİFT KAYNAK KISITLI HÜCRESEL ÜRETİM SİSTEMİNDE PARTİ VE
ÇALIŞAN ÇİZELGELEME PROBLEMLERİ İÇİN BÜTÜNSEL BİR
YÖNTEMBİLİM VE UYGULAMASI**

DOKTORA TEZİ

**Ömer Faruk YILMAZ
(507122107)**

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Mehmet Bülent DURMUŞOĞLU

KASIM 2017

İTÜ, Fen Bilimleri Enstitüsü'nün 507122107 numaralı Doktora Öğrencisi Ömer Faruk YILMAZ, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “ÇİFT KAYNAK KISITLI HÜCRESEL ÜRETİM SİSTEMİNDE PARTİ VE ÇALIŞAN ÇİZELGELEME PROBLEMLERİ İÇİN BÜTÜNSSEL BİR YÖNTEMBİLİM VE UYGULAMASI” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Mehmet Bülent DURMUŞOĞLU**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Doç. Dr. Emre ÇEVİKCAN**
İstanbul Teknik Üniversitesi

Doç. Dr. Şule İTİR SATOĞLU
İstanbul Teknik Üniversitesi

Doç. Dr. Bahadır GÜLSÜN
Yıldız Teknik Üniversitesi

Prof. Dr. Semih ÖNÜT
Yıldız Teknik Üniversitesi

Teslim Tarihi : **3.11.2017**
Savunma Tarihi : **29.11.2017**

Aileme,

ÖNSÖZ

Günümüzde üretim işletmelerinin birçoğu kısıtlı ekipman ve/veya insan kaynaklarına sahiptir. Eğer bir üretim tesisinde hem ekipman hem de insan kaynağı kısıtlı ise tesisin üretim sistemi çift kaynak kısıtlı olarak adlandırılmaktadır. Bu tür üretim sistemlerinde kaynakların kullanımı oldukça önem arz etmektedir. Bu nedenle çizelgeleme ve atama problemlerinin ele alınması ve bu problemler için iyi çözümler geliştirilmesi oldukça önemlidir.

Bu tez kapsamında melez hücrelerden oluşan bir hücresel üretim sisteminde parti salınım ve parti çizelgeleme ve çalışan atama problemleri ele alınmıştır. Hücresel üretim sistemlerinde parti çizelgeleme ve çalışan atama problemi oldukça önemli bir problem türüdür. Kaynakların verimli kullanılması ve etkin çizelgeleme gerçekleştirebilmek açısından uygulanabilir yöntembilimlere ihtiyaç bulunmaktadır. Bu nedenle doktora tezi kapsamında uygulanabilir bir yöntembilim önerilmiştir. Yöntembilimin uygulaması Ayvaz A.Ş. üretim tesisinde gerçekleştirilmiştir.

Doktora ders ve tez dönemim boyunca bana desteğini esirgemeyen, tüm sorularıma sabır ve güleryüzle cevap veren değerli danışman hocam Prof. Dr. Mehmet Bülent Durmuşoğlu' na teşekkürlerimi sunarım.

Kasım 2017

Ömer Faruk YILMAZ
(Endüstri Mühendisi)

İÇİNDEKİLER

Sayfa

ÖNSÖZ	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET	xvii
SUMMARY	xix
1. GİRİŞ	1
2. YAYIN TARAMASI.....	7
2.1 Sipariş (Parti) Salınım Çalışmaları.....	7
2.2 Hücresel Üretim Sisteminde Parti Çizelgeleme Çalışmaları.....	9
2.3 Hücresel Üretim Sistemlerinde Çalışan Tahsisi/Ataması Çalışmaları.....	10
3. ÖNERİLEN YÖNTEMBİLİM	15
3.1 Parti Salınım Problemi	16
3.2 Parti Çizelgeleme Problemi.....	19
3.2.1 Problemin tanımlanması	20
3.2.2 Optimizasyon modeli	24
3.2.3 Problemin NP-Zor yapısı	29
3.2.4 Problemin yapısal özellikleri, alt ve üst sınırlar.....	31
3.2.4.1 Yapısal özellikler	31
3.2.4.2 Alt ve üst sınırlar.....	34
4. META-SEZGİSEL ALGORİTMALAR.....	37
4.1 PÇÇA Probleminin Kodlanması ve Örnek Bir Uygulama.....	37
4.1.1 PÇÇA probleminin çözümleme prosedürü ve amaç fonksiyonu	39
4.2 Genetik Algoritma ve Öncelik Kuralları	41
4.2.1 Genetik algoritma operatörleri	45
5. DENEYSEL HESAPLAMALAR	49
5.1 Deneysel Veriler.....	51
5.2 Değerlendirme Kriteri	53
5.3 Parametre Değerlerinin Hesaplanması	53
5.4 Karşılaştırma Analizi.....	58
6. ÖNERİLEN YÖNTEMBİLİM İLE GERÇEK BİR UYGULAMA.....	73
7. SONUÇLAR VE DEĞERLENDİRMELER	81
KAYNAKLAR	87
EKLER.....	93
ÖZGEÇMİŞ.....	107

KISALTMALAR

AL	: Aktivite Listesi
BVZA	: Bulanık Veri Zarflama Analizi
ÇDKO	: Çalışan Değer Katma Oranı
ÇKK	: Çift Kaynak Kısıtlı
ÇS	: Çevrim Süresi
DF	: Döner Flanşlı
GA	: Genetik Algoritma
GARS	: Genetik Algoritma Rassal Sıralama
GAKİSÖ	: Genetik Algoritma Kısa İşlenme Süresi Önce
GAUİSÖ	: Genetik Algoritma Uzun İşlenme Süresi Önce
GAHBKİSÖ	: Genetik Algoritma Hücre Bazlı Kısa İşlenme Süresi Önce
GAHBUİSÖ	: Genetik Algoritma Hücre Bazlı Uzun İşlenme Süresi Önce
GT	: Grup Teknolojisi
GYS	: Göreli Yüzde Sapması
HBKİSÖ	: Hücre Bazlı Kısa İşlenme Süresi Önce
HBUİSÖ	: Hücre Bazlı Uzun İşlenme Süresi Önce
HPS	: Hücre Parti Sayısı
HMS	: Hücre Makine Sayısı
HS	: Hücre Sayısı
HÜS	: Hücresel Üretim Sistemi
İÜTS	: İlk Ürün Temin Süresi
KÇİ	: Kolbe Çaba İndeksi
KİSÖ	: Kısa İşlenme Süresi Önce
KKPMC	: Kaynak Kısıtlı Paralel Makineler Çizelgeleme
MÜS	: Melez Üretim Sistemi
PB	: Parti Büyüklüğü
PÇÇA	: Parti Çizelgeleme ve Çalışan Atama
PL	: Parti Listesi
PMEKÇ	: Paralel Makineler Esnek Kaynak Çizelgeleme
PS	: Parti Salınım
PSO	: Parçacık Sürü Optimizasyonu
RS	: Rassal Sıralama
SÇŞ	: Seri Çizelge Şeması
SF	: Sabit Flanşlı
SS	: Sipariş Salınım
TÇS	: Toplam Çalışan Sayısı
UİSÖ	: Uzun İşlenme Süresi Önce
YBS	: Yapay Bağışıklık Sistemi

ÇİZELGE LİSTESİ

Sayfa

Çizelge 4.1 : Örnek problem verileri.	38
Çizelge 4.2 : Maksimum ve minimum çalışan sayıları.....	38
Çizelge 4.3 : Parti listesi kodlama yöntemi.	39
Çizelge 4.4 : Hücre çevrim süreleri ve ilk parça temin süreleri.	39
Çizelge 5.1 : Hücre çevrim süreleri, parti büyüklükleri, operasyon süreleri ve toplam yürüme süreleri.	50
Çizelge 5.2 : Küçük boyutlu problemler için faktörler ve seviyeleri.	52
Çizelge 5.3 : Büyük boyutlu problemler için faktörler ve seviyeleri.	52
Çizelge 5.4 : Çalışan yetenek seviyeleri	52
Çizelge 5.5 : Yetenek seviyelerinin katsayıları	52
Çizelge 5.6 : Bilinen en iyi parametre değerleri.	55
Çizelge 5.7: Küçük boyutlu problemler için elde edilen sonuçlar.....	59
Çizelge 5.8: Küçük boyutlu problemler için algoritma GYS(%) değerleri.	60
Çizelge 5.9 : Büyük boyutlu problemler için algoritma GYS(%) değerleri süreleri.	61
Çizelge 5.10 : Çalışan havuzlarının maliyetleri.....	71
Çizelge 6.1 : Üç farklı havuz ve önerilen yöntem bilim uygulama sonuçları	76
Çizelge 6.2 : Çalışan havuzları özet değerlendirme	76

ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 3.1 : Önerilen yöntembilim.....	15
Şekil 3.2 : Melez üretim hücresinde uygulanabilecek akış tipleri.....	21
Şekil 4.1 : Problemin çözümlenmesi ve örnek bir çizelge.....	40
Şekil 4.2 : Örnek problem için melez hücre yapısı.....	41
Şekil 4.3 : Genetik algoritma akış şeması.....	44
Şekil 4.4 : Tek noktalı çaprazlama operatörü	46
Şekil 5.1 : Populasyon parametresinin GYS üzerine etkisi	56
Şekil 5.2 : Mutasyon parametresinin GYS üzerine etkisi.....	56
Şekil 5.3 : Çaprazlama parametresinin GYS üzerine etkisi.....	57
Şekil 5.4 : Algoritmaların GYS değerlerinin 95% güven aralıkları	65
Şekil 5.5 : Algoritmaların hücre sayısı (HS) faktörü üzerinden performansları.....	66
Şekil 5.6 : Algoritmaların hücre parti sayısı (HPS) üzerinden performansları.....	67
Şekil 5.7 : Algoritmaların hücre makine sayısı (HMS) üzerinden performansları....	68
Şekil 5.8 : Algoritmaların toplam çalışan sayısı (TÇS) üzerinden performansları ...	69
Şekil 6.1 : Körük üretim hücreleri	74
Şekil 6.2 : Çalışan değer katma oranlarının (ÇDKO) havuzlara göre değişimi	77
Şekil 6.3 : Önerilen yöntembilim uygulaması ile elde edilen çizelge	78

ÇİFT KAYNAK KISITLI HÜCRESEL ÜRETİM SİSTEMİNDE PARTİ VE ÇALIŞAN ÇİZELGELEME PROBLEMLERİ İÇİN BÜTÜNSEL BİR YÖNTEMBİLİM VE UYGULAMASI

ÖZET

Hem manuel hem de otomatik operasyonların gerçekleştirildiği bir hücresel üretim sisteminde (HÜS) bulunan hücreler melez olarak adlandırılmaktadır. Bu tür hücrelerde hem manuel hem de otomatik operasyonlar gerçekleştirilebildiğinden hücre çevrim süreleri değişkenlik göstermektedir.

Bu tez kapsamında çift kaynak kısıtlı (ÇKK) melez üretim hücrelerinden oluşan hücresel üretim sisteminde (HÜS) parti salınım (PS) ve parti çizelgeleme ve çalışan atama (PÇÇA) problemleri ele alınmıştır. PS probleminin çözülmesiyle işyükü kontrolü sağlanırken PÇÇA probleminin çözülmesi ile birlikte işyükünün çizelgelenmesi sağlanmaktadır. İşyükü çizelgelenmesinden kasıt çalışanların ve partilerin eş zamanlı olarak çizelgelenmesidir. Ele alınan sistemde periyodik salınım gerçekleştirilmekte ve belirli bir zaman kısıtı altında partilerin sisteme salınımı sağlanmaktadır. Aşırı miktarda partinin sisteme salınması olurlu çizelgelerin elde edilmesini önlerken düşük miktarda salınım ise verimsizliklere neden olmaktadır. Bu nedenle uygun salınım miktarlarının belirlenmesi gerekmektedir. Sisteme salınan partilerin belirli bir zaman aralığında çalışan atamalarının da gerçekleştirilerek çizelgelenmesi için PÇÇA problemi tanıtılmıştır. Bu problemin çözülmesi ile birlikte partilerin hücrelerdeki operasyonlarına tüm zaman aralıkları için hangi çalışanların atandıkları belirlenebilmektedir.

Yukarıdaki bilgiler değerlendirildiğinde hem sisteme salınan partilerin belirlenmesi ve işyükü kontrolü için ortaya koyulan PS probleminin hem de salınan partilerin çalışanlarla birlikte çizelgelenmesi için ortaya koyulan PÇÇA problemlerinin birbirleriyle ilişkili oldukları görülmektedir. Bu nedenle bu tez kapsamında her iki problem geliştirilmiş olan bir yöntembilim altında aşamalı olarak birleştirilmiştir. Yöntembilim kullanılarak ilk olarak PS problemine çözüm aranmakta ardından PS problemi için bulunan çözüm PÇÇA problemi için girdi olarak kullanılarak probleme çözüm aranmaktadır. Her iki problemin net olarak ifade edilmesi ve çözüm aranması amacıyla matematiksel modeller geliştirilmiştir. PS problemi için geliştirilen matematiksel modelde amaç sisteme salınan parça sayısının en büyüklenmesi olarak belirlenirken PÇÇA problemi için geliştirilen matematiksel modelde amaç toplam çalışan maliyetinin en küçüklenmesidir. Bu sayede hem yayılma süresi hem de çalışan maliyetleri eş zamanlı olarak düşürülebilmektedir.

Önerilen yöntembilimin uygulanması için Ayvaz A.Ş. körük üretim tesisi seçilmiştir. Bu üretim tesisinin birbirinden bağımsız birden çok melez üretim hücresinden oluştuğu ve çift kaynak kısıtlı bir sistem olduğu gözlemlenmiştir. Sistemin ÇKK olmasının nedeni çıktı sayılarının hem hücrelere atanan çalışanların kapasitelerine hem de mevcut hücre kapasitelerine bağımlı olmasıdır. Bu nedenle sistem çift kaynak kısıtlı hücresel üretim sistemi (ÇKK HÜS) olarak adlandırılmıştır. Melez üretim hücrelerinde hem manuel hem de otomatik operasyonlar

gerçekleştirilebilmektedir. Hücrelerdeki operasyonlara atanan çalışanların yetenek seviyeleri ve sayıları hücre çevrim süresini direkt olarak etkilemektedir. Böylece parti işlem süreleri ve hücre çıktı miktarları dolaylı olarak etkilenmektedir. Bu etkiler dikkate alınarak tek parça akışa, çalışan yetenek seviyelerine ve çalışan uygunluklarına PÇÇA problemi için geliştirilen matematiksel modelde yer verilmiştir.

PÇÇA problemine çözüm bulabilmek için genetik algoritma (GA) ile birlikte beş farklı öncelik kuralı kullanılmış ve her öncelik kuralı farklı bir algoritma olarak değerlendirilerek küçük ve büyük boyutlu problemler üzerinden karşılaştırılmaları sağlanmıştır. Küçük boyutlu problemler için tüm algoritmalar benzer sonuçlar sağlarken büyük boyutlu problemlerde en iyi sonuçların genetik algoritma rassal sıralama algoritması tarafından sağlandığı gözlemlenmiştir.

Önerilen yöntemimizin AYVAZ A.Ş. üretim tesisinde uygulanması ile birlikte performans göstergesi olarak belirlenen çalışan değer katma oranının (ÇDKO) iyileştiği gözlemlenmiştir.

A HOLISTIC METHODOLOGY AND APPLICATION FOR BATCH AND WORKER SCHEDULING PROBLEMS IN A DUAL RESOURCE CONSTRAINED CELLULAR MANUFACTURING SYSTEM

SUMMARY

Cellular manufacturing systems (CMS) are effective production systems for producing part families, which are composed of parts with similar processing requirements. Through the utilization of CMS, the setup time, work-in-process inventories, material handling cost, and lead time are reduced, while the visibility and quality are enhanced. There are several activities to be carried out in CMS before starting the production process. Scheduling is the final operational activity, and plays a key role in the smooth operation of CMS. Hence, finding good solutions to the scheduling problems is very important for the successful implementation of CMS.

Scheduling problems in cellular manufacturing system (CMS) are widely investigated in the academic literature to increase the operational efficiency in production. These problems can mainly be classified into two groups: those that consider a single cell and those that consider multiple cells.

The process for cell scheduling, known as group scheduling, can be divided into family and job scheduling stages. The family sequence and the job sequences at each family are determined simultaneously in order to compute the completion time for each job. However, the effect of employee skills on the cell cycle times, employee timetabling, and the application of one-piece flow within cells are not considered in cell scheduling. For this reason, a comprehensive batch scheduling problem that integrates employee timetabling and cell scheduling by considering employee skills together with the application of one-piece flow within cells would be a valuable addition to the existing academic literature.

In the current thesis, the batch scheduling problem in a cellular manufacturing system (CMS) is addressed, considering skilled workforce assignment (SWA), which refers to employee flexibility. The CMS consists of a set of independent and parallel hybrid manufacturing cells. Unlike automated and assembly cells, both automatic and manual operations are performed in these hybrid cells, hence the reason for calling the cells hybrid.

The importance of SWA for batch scheduling problems becomes especially clear for the CMS. As long as the employee flexibility is concerned for these cells, it requires taking into account the different skill levels for the employees to perform the manual operations; thus, the SWA has a crucial effect on the CMS performance whenever the manual operations have to be performed in the cells.

Production systems in a dual-resource constrained (DRC) setting are the ones in which the capacity constraints on the output depend on the availability of both the machines and the employees. Employees required for manual operations are considered to be valuable resources when there are fewer employees than the number

of workstations at which manual operations are performed. This kind of cellular manufacturing system (CMS) is considered to be a DRC system, which is a well-established trend in the literature.

In this type of system, human operators are often constrained resources, and the movements of employees are both inter-cellular and intra-cellular. One of the most notable examples of CMS in a DRC setting in which employee resources directly affect the processing times of the batches are hybrid manufacturing cells, where both automatic and manual operations are performed. Furthermore, employee involvement is unrestricted in the hybrid cells, and the assignment of employees has a significant effect on the cell cycle times. Because of the manual operations within the hybrid cells, the cell cycle times vary significantly, and the batch processing times depend on the employees assigned to perform the operations on these batches. Considering employees as the critical resources for both their availability and effect on the manual operation times makes it more important to research this problem. In real manufacturing environments, employees are not unlimited resources, and they cannot be treated equally regarding their skill levels.

In this thesis, the cells are arranged in a multi-product flow shop layout that applies one-piece flow, and the CMS has a DRC setting in such a manner that a batch can only be processed if employee resources and machine resources are available.

Most studies on batch scheduling have focused on static scheduling problems. In these types of problems, employee resources are accepted to be non-renewable, and they cannot be assigned to more than one operation. In addition, the number of employees in each cell remains unchanged during the scheduling horizon. On the other hand, the employee resources are assumed to be renewable for dynamic scheduling problems, so they can be assigned to several non-overlapping operations. Therefore, the number of employees in each cell varies dynamically throughout the scheduling horizon. The current thesis focuses on the dynamic batch scheduling problem, considering employee resources to be limited.

The contributions of this thesis are as follows:

A comprehensive methodology is proposed to handle order release, batch scheduling and worker assignment problems all together.

The batch scheduling problem considering one-piece flow application within cells for the CMS is investigated, which has not been studied in the academic literature to date. In addition, SWA with a DRC setting for CMS is considered. A new optimization model and holistic methodology are presented for the problem. A lower bound is determined to use for an experimental comparison of the meta-heuristics.

Several structural properties are described, for a better explanation of the problem. The metaheuristics are compared regarding the solution quality, and the best among them is determined as the result of detailed analyses.

In this thesis, the batch scheduling problem for CMS in a DRC setting is examined, considering SWA. The decision problems to be met include determining the sequence and starting times of batches, and assigning employees to cells in which operations are performed while considering employee skill levels.

The development of an integer linear optimization model is described for the problem. Because the considered problem is NP-hard in the strong sense, five metaheuristics have been proposed, and compared with each other. For comparison

purposes, computational experiments were conducted using a set of test instances that were generated based on real production data.

The proposed methodology was applied on a real manufacturing system and the improvements were shown in this thesis.

1. GİRİŞ

Grup teknolojisi (GT), parçalar/ürünler arasındaki yapısal ve işlemsel benzerlikleri temel alarak ürünlerin sınıflandırılmasına dayalı ve parti üretimine izin veren bir üretim felsefesidir (Won ve Currie, 2004). Dört temel prensibi bulunmaktadır: sınıflandırma, aile oluşturma, basitleştirme, standartlaştırma (Ham ve diğ, 1985).

Hücresele üretim sistemleri (HÜS) grup teknolojisi ilkelerinin üretim sistemlerine uygulanmasıdır (Rajesh ve diğ, 2017). Bu sistemlerde benzer işlem gereksinimlerine sahip parçalar ve bu parçaların işlenmesi için gerekli makineler belirlenerek parçamakine aileleri oluşturulmakta ve her parça ailesi kendisi için tahsis edilen bir hücreye atanmaktadır (Venkataramanaiah, 2008; Hachicha ve diğ, 2007). Bu sayede HÜS birden fazla birbirine en az düzeyde bağımlı (mümkünse bağımsız) hücrelerden oluşan bir üretim sistemi olarak tanımlanmaktadır. Hücrelerin birbirinden bağımsız olması için hücreler arası hareket eden istisnai parçaların olmaması veya yok edilmesi gerekmektedir. İstisnai parçaların bir hücrede üretilen parçaların bir başka hücredeki herhangi bir operasyona/makineye ihtiyaç duyması sonucunda oluşmaktadır. Eğer her hücre kendisine tahsis edilmiş parça ailesini üretebilecek yeterli kapasitede makinelerin tümüne sahipse bu durumda istisnai elemanlar oluşmamaktadır. İstisnai elemanlar ne kadar az sayıda ise HÜS işleyişi o derecede tek parça akışına imkan sağlamaktadır. Her hücre tek parça akışını istisnai elemanlar olmadan sağlayabildiğinde ideal bir HÜS'den bahsedilebilir. Bunun sağlanması için HÜS tasarımı oldukça önem arz etmektedir.

Hücresele üretim sistemlerinde, ürün yaşam çevriminin orta düzeylerinde bulunan ve düzgün talepli parçaların üretimi yapılmaktadır. Bu parçalar göz önünde bulundurularak HÜS tasarımına karar verilmektedir. Literatürde HÜS tasarımı ile ilgili fazla sayıda çalışma bulunmaktadır (Yin ve Yasuda 2005; Yin ve Yasuda 2006; Boe ve Cheng 1991; Hamza ve Adesta 2013; King 1980; Chandrasekharan ve Rajagopalan, 1986). Bu tez kapsamında HÜS tasarımına yer verilmediği için ilgilenen okuyucular bu çalışmalardan faydalanabilir.

Eğer bir üretim sisteminde ürün yaşam çevriminin hem olgunluk aşamasında hem de ilk veya son aşamasında yani dengesiz talebe sahip parçalar üretiliyorsa bu durumda melez üretim sistemlerinin (MÜS) tasarlanması daha uygundur. Literatürde MÜS tasarımı ve operasyonları ile ilgili olarak birçok çalışma bulunmaktadır. (Yılmaz ve Durmuşoğlu, 2017; Durmuşoğlu ve Satoğlu, 2011; Ağlan ve Durmuşoğlu, 2015). İlgilenen okuyucular bu çalışmalardan yararlanabilir.

HÜS'nin diğer üretim sistemlerine nazaran birçok avantajı bulunmaktadır. Avantajlar şu şekilde sıralanabilir: üretim temin süresinde azalış, çalışan ilişkilerinin iyileşmesi, hazırlık süresinde azalış, süreç içi stoklarda azalış, görsellikte ve kalitede artış, malzeme elleçlemede verimlilik, esneklik artışı ve parça/parti ve çalışan çizelgelemede kolaylık (Yılmaz ve diğ., 2016). Yer verilen avantajların detaylı açıklamasına aşağıda yer verilmektedir.

HÜS'nin tasarımı yapıldıktan sonra her hücrede tek parça akışın sağlanması gerekmektedir. Tek parça akış sağlandıktan sonra üretim temin süresinde ciddi oranlarda azalış meydana gelmektedir. Tek parça akışın üretim temin süresi üzerine olan etkisine kısım 3.2.1' de yer verilmektedir.

HÜS'nin içerisinde operasyonlar bir veya birden fazla çok fonksiyonel çalışan tarafından gerçekleştirildiği için HÜS'leri çalışan ilişkilerinin iyileşmesine katkı sağlar. Askin ve Huang (2001), etkin çalışan gruplarının oluşturulması için Kolbe (1993) tarafından geliştirilen Kolbe Çaba İndeksini (KÇİ) kullanarak hücrelerin oluşturulması aşamasında çalışan ilişkilerini de göz önünde bulundurmuşlardır. İlgilenen okuyucular Askin ve Huang (2001) çalışmasından faydalanabilirler.

Parça-makine aileleri oluşturulduğu ve parça aileleri ayrı hücrelere tahsis edildiği için hazırlık sürelerinde ciddi anlamda azalış meydana gelmektedir.

Tek parça akışın getirmiş olduğu bir diğer avantaj ise süreç içi stokların azalmasıdır. HÜS'de yalın üretim tekniklerinin uygulanması ile birlikte sipariş için yapılan üretim tek parça akış ile desteklendiğinde süreç içi stoklar ciddi anlamda azalmaktadır.

Görsellik yalın üretim teknikleri içerisinde oldukça önemlidir. Tüm hattın görsel olarak izlenebilmesi ve hata anında durdurulması önem arz etmektedir. Ayrıca görsel panoların hazırlanarak tüm çalışanların süreci bir bütün olarak görebilmesi ve katılım göstermesi önemlidir. Hücresel üretim sistemleri yalın üretime giden yolda önemli

bir aşamadır ve göreselliğin sağlanmasına olanak sağlamaktadırlar. Ayrıca tek parça akışı ile birlikte hata anında hatların ve hücrelerin durdurulması ve işlenen parçaya odaklanılabilmesi sayesinde insan kaynaklı kalite sorunları azaltılabilmektedir. Unutulmamalıdır ki HÜS'leri yalın üretim yolunda önemli bir aşamadır ancak Kaizen, 5S ve diğer yalın üretim teknikleri ile desteklendiklerinde avantajları daha fazla ön plana çıkmaktadır.

Malzeme elleçlemedeki verimliliğin nedeni partiler yerine parti içindeki ürünlerin elleçlenmesidir. Hücresel üretim sistemleri sayesinde hem çalışan hem de parça esnekliği sağlanmaktadır.

Çalışanlar çok fonksiyonel olduklarında hücreler arası transfere açık olabilmektedirler. Böylece çalışan esnekliği sağlanmaktadır. Bir parça ailesine tahsis edilmiş bir hücre aynı parça ailesinden birden fazla parça işleyebildiği için parça esnekliği sağlanmış olmaktadır.

HÜS'leri fonksiyonel üretim ile karşılaştırıldıklarında çizelgeleme konularında ciddi anlamda kolaylık sağlamaktadırlar. Bir HÜS birbirine paralel birden çok hücreden meydana gelmektedir ve çizelgeleme yapıldıktan sonra hangi partinin hangi çalışan ile birlikte ne zaman işlenmeye başlanacağı belirlenebilmektedir. Böylece üretim operasyonlarının takibi daha rahat yürütülebilmektedir. Fonksiyonel üretimde birbirine benzer makineler bir arada bulunduğu için ürünler makine gruplarına göre işlenmektedirler. Bu durum her makine için hangi ürünün ne zaman işlenmeye başlanacağını belirlenmesini gerektirir. Böylece üretim operasyonlarının takibi ve yürütülmesi daha zordur ve uzun zaman alır. Ayrıca fonksiyonel üretimde genellikle parti üretimi yapılmakta ve makine grupları arasında parti dolaşımı olmaktadır. Bu durum üretim temin süresinin büyük oranda artmasına neden olur.

Bu tez kapsamında hücresel üretim sisteminde parti salınım ve parti ve çalışan çizelgeleme problemlerini birleştirerek ortak bir çözüm bulmaya olanak sağlayan iki aşamalı bütünsel bir yöntem bilim önerilmektedir. Önerilen yöntem bilimin ilk aşamasında parti salınım problemine en iyi çözüm bulunmakta ve ikinci aşamada parti ve çalışan çizelgeleme problemleri birlikte ele alınarak geliştirilen meta-sezgisel yöntemlerle ortak çözüm aranmaktadır. İlk aşama sistemdeki işyükünün kontrol altında tutulmasını sağlamaktadır ve ikinci aşamada ilk aşamanın sonuçları girdi parametresi olarak kullanılarak sisteme üretilmek üzere salınan partiler mevcut

alışan havuzundan yararlanılarak izelgelenmektedir. Bylece hem parti hem de alışan izelgelenmesi gerekleřtirilebilmektedir.

İlk ařamadaki problem polinomsal zamanda zlebilen P sınıfı bir problem olduėundan dolayı en iyi zme GAMS[®] 23.5/CPLEX 12.2 paket programı kullanarak ulařılabilmektedir. Ancak ikinci ařamada yer verilen problem NP-Zor sınıfında bir problem olduėundan meta-sezgisel bir yntem olan genetik algoritma (GA) beř farklı ncelik kuralı ile ayrı ayrı alıřtırılarak problemin zmnde kullanılmıřtır. Bu sayede, beř farklı ncelik kuralı aynı zamanda beř farklı algoritma olarak deėerlendirilmiř ve sezgisel algoritmalar birbirleriyle zm kalitesi ve sresi bakımından karřılařtırılmıřtır. Karřılařtırma yapabilmek iin kk ve byk boyutlu problemler oluřturulmuřtur. Kk boyutlu problemler iin hem GAMS[®] 23.5/CPLEX 12.2 paket programı kullanarak en iyi zmler bulunmuř hem de algoritmalarından olurlu zmler saėlanmıřtır. Byk boyutlu problemler iin alt sınır ve algoritmalar tarafından saėlanan en iyi zm deėerleri karřılařtırma amalı kullanılmıřtır. Her boyut problem sınıfı iin farklı hcre sayısı, parti sayısı, makine sayısı ve alışan havuzları kullanarak farklı problemler retilmiřtir. alışan havuzları retilirken alışanların farklı yetenek seviyeleri gz nnde bulundurulmuř ve farklı yeteneklerdeki alışanlardan oluřan havuzların problemin zmne olan katkısı detaylı olarak incelenmiřtir.

Algoritmalar kendi aralarında karřılařtırılmadan nce girdi parametrelerinin doėru belirlenmesi performans aısından olduka nemlidir. Parametrelerin belirlenebilmesi amacıyla deneme yanılma yntemiyle tekrarlı denemeler gerekleřtirilmiř kullanılan parametre deėerleri arasından en iyi performans saėlayan parametre kombinasyonları kullanılmıřtır. Parametre kestirimi ve deneysel sonular kısım 5.3' de detaylı olarak aıklanmıřtır.

Parti ve alışan izelgeleme problemi iin kullanılan algoritmalar arasından en iyi sonucu veren algoritma nerilen yntembilimin ikinci ařamasında kullanarak gerek retim ortamından saėlanan veriler zerinden yntembilimin uygulaması gerekleřtirilmiřtir. Uygulama gerekleřtirilirken farklı alışan havuz stratejileri tek vardiya dikkate alınarak deėerlendirilmiřtir. Detaylı uygulama sonularına kısım 5.4' de yer verilmiřtir.

Tezin ikinci kısmında sipariş (parti) salınım ve HÜS’de çizelgeleme ve çalışan atama problemleri ile ilgili yayın taramasına yer verilmektedir. Önerilen yöntem bilim üçüncü kısımda açıklanmaktadır. Üçüncü kısım içerisinde, önerilen yöntem bilimin birinci aşaması için geliştirilen parti salınım matematiksel modeline ve ikinci aşaması için geliştirilen parti ve çalışan çizelgeleme matematiksel modeline yer verilmektedir. Aynı kısım içerisinde problemlerin açıklanması, problemin özellikleri için matematiksel ispatlara ve problem için geliştirilen alt sınır denklemine yer verilmektedir. Tezin dördüncü kısmında parti ve çalışan çizelgeleme problemi için kullanılan genetik algoritma ve beş farklı öncelik kuralına yer verilmektedir. Aynı kısım içerisinde uygun parametre kombinasyonları belirlemek için kullanılan deneme yanılma yöntemine ve karşılaştırma sonuçlarına yer verilmektedir. Tezin beşinci kısmında gerçek uygulama ve çizelgeleme sonuçlarına yer verilmektedir. Tezin son ve altıncı kısmında ise sonuçlar, değerlendirmeler ve gelecek çalışmalar için önerilere yer verilmektedir.

2. YAYIN TARAMASI

Bu tez kapsamında hücresele üretim sistemlerinde parti salınımı, parti ve çalışan çizelgeleme problemlerine yer verildiği için ilgili problemlerle ilgili literatürde bulunan çalışmalara bu kısımda yer verilmektedir. İlk alt başlık altında sipariş salınım (SS) problemi ile ilgili çalışmalara yer verilmekte, ikinci alt başlık altında hücresele üretim sistemlerinde sıralama ve çizelgeleme çalışmalarına ve son olarak üçüncü alt başlık altında hücresele üretim sistemlerinde çalışan atama ve çizelgeleme çalışmalarına yer verilmektedir.

2.1 Sipariş (Parti) Salınım Çalışmaları

Sipariş salınım (SS) üretim kontrol politikası işyükü kontrolünden bir önceki adım olabileceği gibi işyükü kontrolünün ana adımlarından bir tanesi de olabilir. İki farklı SS politikası bulunmaktadır: Periyodik Salınım ve Sürekli Salınım politikaları. SS ile ilgili bu iki politikayı incelemek amacıyla literatürde benzetim tekniğini kullanan çalışmaların yanısıra matematiksel modellerin kullanıldığı çalışmalarda bulunmaktadır. Bu çalışmaların birçoğu fonksiyonel veya akış tipi üretim sistemleri için gerçekleştirilmiştir. Bu kısımda HÜS için gerçekleştirilen SS çalışmalarına yer verilmektedir.

Mahmoodi ve diğ. (1990) üretim sistemi karakteristiklerinin SS için öneminden bahsetmiş ve SS mekanizmasının buna bağlı olarak geliştirilmesi gerektiğini belirtmişlerdir. Çalışmalarında üç farklı SS ve iki farklı teslim zamanı belirleme sezgisellerinin kombinasyonları ile oluşturdukları toplam altı farklı sezgisel yöntem kullanmışlardır. Çalışmanın sonuçlarında kontrollü periyodik salınımın akış süresi ve gecikme performansı üzerinde etkili olduğu vurgulanmıştır. Batsis ve Tatsiopoulos (1993) SS için oluşturmuş oldukları havuz sistemini malzeme ihtiyaç planlamasından gelen üretim emirleri ile çalıştırmışlardır. Havuz sisteminden üretim için hangi ürünün salınacağına bir girdi kontrol mekanizması ile karar vermişlerdir. Ashby ve Uzsoy (1995) sipariş teslim zamanları ve sıra bağımlı hazırlık sürelerini dikkate alarak geliştirdikleri çizelgeleme sezgiselleri ile birlikte periyodik SS ve grup

izelgeleme problemlerine özüm aramışlardır. alışmanın sonuçlarında hazırlık süresi, teslim zamanları ve öncelik kurallarının sistem performansı üzerinde ciddi etkileri olduğunu belirtmişlerdir. Quintana (2002) paraların tek para akışın uygulandığı hücresele üretim ortamına salınımı için melez bir mekanizma geliştirmiş ve bu mekanizmanın kontrolü için bir algoritma tasarlamışlardır. Önermiş olduğu yöntembilimi benzetim tekniğı ile deęerlendirmiştir. alışmanın sonuçlarında uygulanan SS stratejisinin zamanında tamamlanan sipariş sayısı üzerinde oldukça etkili olduğunu vurgulamıştır. Mönch (2005) farklı SS stratejilerinin geliştirmiş oldukları bir sezgisel üzerindeki etkilerini incelemişlerdir. Önerdikleri sezgiselin SS üzerindeki performansının öncelik kurallarından daha iyi olduğunu belirtmişlerdir.

Hücresele üretim sistemlerinde sipariş salınım konusunda gerçekleştirilen alışmalara yukarıda yer verilmiştir. Bu alışmalardan aşağıdaki sonuçlar ıkarılmıştır:

- Uygulanan SS metodunun sistemin akış süresi, gecikme zamanı, gecikme sayısı, sipariş tamamlanma oranı v.b performans göstergeleri üzerindeki etkisini incelemek amacıyla benzetim metodunun yanısıra sezgisellerden elde edilen deneysel sonuçlar da kullanılmaktadır.
- SS politikası olarak öncelik kuralları tek başına kullanılabileceğı gibi tamamlanma zamanı belirleme sezgiselleri ile ortak olarak da kullanılabılırler. SS politikası olarak problem spesifik sezgiseller geliştirilmiş ve öncelik kuralları ile karşılaştırıldıklarında daha iyi sonuçlar verdikleri gözlemlenmiştir.
- alışmaların birçoğunda SS problemi tek başına ele alınmamış aynı zamanda hücre izelgeleme (grup izelgeleme) problemi ile birlikte ele alınmıştır. Bu nedenle en iyi sonucun bulunması için matematiksel modellerden yararlanmak mümkün olmamıştır.
- Üretim ortamı karakteristiklerinin SS metodunun belirlenmesinde önemli olduğu gözlemlenmiştir.

Bu tez kapsamında SS problemi parti izelgelemeden bağımsız olarak deęerlendirilmiştir. Yukarıda da belirtildiğı üzere SS, işyükü kontrolünden önceki adım olabileceğı gibi iş yükü kontrolün ana bir adımı da olabilmektedir. Önerilen bütünsel yöntembilimin ilk aşamasında periyodik salınım sağlamak amacıyla SS problemi ele alınarak polinomsal süreler içerisinde problemin en iyi özümüne ulaşılmıştır. Elde edilen ıktılar ikinci aşamada girdi olarak kullanılmıştır. Böylece

bu tez kapsamında ele alınan parti salınım matematiksel modeli işyükü kontrolünün ana adımlarından bir tanesi olarak değerlendirilmektedir.

2.2 Hücresel Üretim Sistemlerinde Parti Çizelgeleme Çalışmaları

Hücresel üretim sistemlerinde operasyonel etkinliği arttırmak amacıyla çizelgeleme çalışmaları önem arz etmektedir. Tek hücreli sistemleri ele alan çizelgeleme çalışmalarının yanı sıra çok hücreli sistemleri ele alan çalışmalarda bulunmaktadır (Li ve diğ., 2015). Her iki sınıf çalışmalarda genellikle grup çizelgeleme problemine odaklanılmaktadır. Bu problem iki alt kısma ayrılmaktadır. İlk olarak hücreye atanan ürün ailelerinin sırası belirlenmekte daha sonra her ürün ailesinin içerisindeki ürünlerin sıraları belirli bir amacı en iyilemek üzere belirlenmektedir (Schaller ve diğ., 2000). Grup çizelgeleme çalışmaları akış tipi hücreler üzerinde gerçekleştirildiği gibi sipariş tipi hücrelerde de gerçekleştirilmektedir. Grup çizelgeleme problemlerine odaklanılan çalışmalarda parti çizelgeleme problemi ele alınmamaktadır. Dolayısıyla çalışan çizelgeleme problemine de yer verilmemektedir. Bu tez kapsamında grup çizelgeleme problemine yer verilmemektedir. Ancak ilgililenen okuyucular Chen ve Lee (2001), Gupta ve Schaller (2005), Venkataramanaiah (2008), Solimanpur ve Elmi (2011) çalışmalarından yararlanabilirler.

Literatürde hücresel üretim sistemlerinde parti çizelgeleme problemini ele alan çalışma sayısı azdır. Aşağıda bu çalışmalara yer verilmektedir.

Das ve Canel (2005) akış tipi hücrelerden oluşan bir esnek üretim sisteminde parti çizelgeleme problemini ele almışlardır. Geliştirmiş oldukları dal sınır algoritmasının büyük boyutlu problemleri kabul edilebilir zamanlarda çözdüğünü göstermişlerdir. Çalışmalarında hücre çevrim süresinin parti işlenme süresi üzerindeki etkisini ve çalışan çizelgeleme problemini ele almamışlardır. Celano ve diğ. (2008) birbirinden bağımsız paralel ve sadece otomatik operasyonlar gerçekleştirilen CNC üretim hücrelerinden oluşan kısıtlı sayıda çalışana sahip bir hücresel üretim sisteminde parti çizelgeleme problemini ele almışlardır. Çalışmalarında amaç olarak sistemdeki çalışan sayısının makine sayısından daha az olmasını sağlayarak aynı miktarda çıktının elde edilmesidir. Farklı üretim senaryoları altında elde ettikleri benzetim sonuçları etkin bir planlama ile çalışan sayısı hücre sayısından küçük olsa bile aynı miktarda çıktı elde edilebileceğini göstermektedir. Çalışmalarında manuel operasyonları ele almamış ve hücre çevrim süresi kavramını kullanmamışlardır.

Balaji ve Porselvi (2014) sıra bağımlı hazırlık sürelerinin olduğu akış tipi hücrelerden oluşan bir hücresel üretim sisteminde parti çizelgeleme problemini ele almışlardır. Yayılma süresini en küçükleme amacıyla problem için bir matematiksel model geliştirmişler ve problem NP-Zor sınıfında olduğu için çözümünde yapay bağımsızlık sistemleri (YBS) ve benzetim tavlama algoritmalarını kullanmışlardır. Çalışmanın sonucunda benzetim tavlama algoritmasının hız açısından daha iyi olduğu ancak YBS algoritmasından elde edilen sonuçların kalitesinin çok daha iyi olduğu sonucuna ulaşmışlardır. Yılmaz ve diğ. (2016) çalışmalarında melez üretim hücrelerinden oluşan bir hücresel üretim sisteminde parti çizelgeleme problemi için hedef programlama modeli geliştirmişlerdir. Matematiksel model de amaç fonksiyonu olarak ortalama akış süresini kullanarak sistemde ki bekleme sürelerinin enazlanmasını amaçlamışlardır. Çalışan sayısı kısıt olarak kullanılmak yerine amaç fonksiyonunda kullanılmış ve geliştirmiş oldukları alt sınır denklemleri vasıtasıyla normalizasyon yaparak farklı amaç fonksiyonu ağırlıklarında farklı değerler elde etmişlerdir. Geliştirmiş oldukları sezgiselleri gerçek üretim verilerinden elde edilmiş problemlerin çözümünde kullanarak etkinliklerini göstermişlerdir.

Literatüre bakıldığında Yılmaz ve diğ. (2016) ve Yılmaz ve Durmuşoğlu (2017) çalışmaları haricinde hücre çevrim süresini ve buna bağlı olarak çalışan çizelgeleme problemini parti çizelgeleme problemi ile bütünleştirerek ele alan herhangi bir çalışma olmadığı görülmektedir. Bu anlamda bu tez ile hücre çevrim sürelerinin değişim gösterdiği parti çizelgeleme ve çalışan atama problemi (PÇÇA) tanıtılacak ve literatürdeki bu boşluk doldurulmuş olacaktır.

2.3 Hücresel Üretim Sistemlerinde Çalışan Tahsisi/Ataması Çalışmaları

Bu tez kapsamında ele alınan problem ile çalışanların partilere bağımlı olarak hücrelere atanması ve bu kapsamda çalışanların partiler ile eş zamanlı olarak çizelgelenmesi amaçlanmaktadır. Bu nedenle bu kısımda HÜS’de çalışan tahsis/atama çalışmalarına yer verilerek literatürde bu probleme nasıl yer verildiği ön plana çıkarılmaya çalışılmıştır.

Russel ve diğ. (1991) çift kaynak kısıtlı (ÇKK) hücresel üretim ortamında benzetim tekniği ile birçok senaryo deneyerek en uygun çalışan tahsis stratejisini belirlemeye çalışmışlardır. Çalışmada ek olarak ürünlerin farklı rotalar izleyebilmesinin ve ürünlerin çizelgeleme politikalarının çalışan tahsisi üzerine olan etkisi incelenmiştir.

Jensen (2000) çalışan esnekliğinin performans avantajlarını farklı hücresel üretim ortamı şartları için benzetim tekniği ile incelemiştir. Çalışmada çalışan transfer kurallarının çalışan kısıtlı hücrelerdeki önemi ve çalışan esnekliği ile çalışan seviyeleri arasındaki önem ispatlanmıştır. Askin ve Huang (2001) çalışanların bireysel olarak hücrede yapacakları görevlere uyumluluğu ve çalışanların etkin takımlar olabilmelerini göz önüne alan karma tamsayılı hedef programlama modeli geliştirmişlerdir. Çalışmalarında etkin çalışan gruplarının oluşturulması için Kolbe (1993) tarafından geliştirilen Kolbe Çaba İndeksi (KÇİ) kullanarak etkin çalışan gruplarının oluşturulmasında kullanmışlardır. Büyük boyutlu problemler için sezgisel algoritmalar önermişlerdir. Norman ve diğ. (2002) organizasyonun verimliliğini arttırmak amacıyla çalışanların üretim hücrelerine atanması problemini ele almışlardır. Çalışmalarında organizasyonun verimliliğini üretkenliğin, çıktı kalitesinin ve eğitim maliyetinin bir fonksiyonu olarak değerlendirmişlerdir. Problemi matematiksel olarak ifade edebilmek adına karma tamsayılı matematiksel model oluşturmuşlardır. Süer ve Dagli (2005) çalışmalarında hücre yükleme ve çalışan tahsisi problemleriyle ilgilenmişlerdir. İlk aşamada hücre içi çalışan transferini en küçükleme için hücreye giren ürünlerin sıralanması problemi ile ilgilenmişlerdir. İkinci aşamada makine-seviye-tabanlı benzerlik katsayıları belirlenmiştir. Üçüncü aşamada ise bu katsayılar kullanarak hücre yüklemeyi gerçekleştirmişlerdir. Cesani ve Steudel (2005) çalışmalarında hücre içerisinde çalışan hareketliliği olarak karakterize edilen çalışan esnekliğine odaklanmışlardır. Farklı çalışan tahsis stratejilerinin sistem performansı üzerine etkilerini benzetim tekniği kullanarak incelemiştir. Sonuçlar göstermiştir ki çalışanların işyükü dengesi sistemin performansını belirleyen önemli bir unsurdur. Süer ve Tummaluri (2008) çalışan yoğun hücrelerde operatörlerin operasyonlara atanma problemi ile ilgilenmişlerdir. Problemin çözümü için 3 aşamalı bir yaklaşım geliştirmişlerdir. İlk olarak alternatif hücre konfigürasyonları belirlenmiş, ikinci olarak hücreler yüklenmiş ve çalışan sayısı belirlenmiş ve üçüncü olarak operatörler operasyonlara atanmıştır. Satoğlu ve Suresh (2009) HÜS içeren bir melez üretim sistem tasarımı için üç aşamadan oluşan bir yaklaşım geliştirmişlerdir. Bu yaklaşımın üçüncü aşamasında çalışan tahsisi için hedef programlama matematiksel modeli geliştirmişlerdir. Uygulama için gerçek bir üretim ortamının verileri kullanılmış ve matematiksel model GAMS optimizasyon programı ile çözülmüştür. Fan ve diğ. (2010) hücre oluşturma ve çalışan atama problemleri için çift kaynak kısıtlı çok

amaçlı bir optimizasyon modeli geliştirmişlerdir. Modeldeki amaçlar, hücreler arası parça ve çalışan hareketlerinin en küçüklenmesi, toplam çalışan sayısının en küçüklenmesi ve çalışan işyükü dengesizliklerinin en küçüklenmesi olarak belirlenmiştir. Modelin çözümü için meta-sezgisel bir yöntem olan genetik algoritma kullanmışlardır. Süer ve Alhawari (2011) çalışan yoğun üretim hücrelerinde iki farklı çalışan tahsis stratejisini (Max-Min ve Max) incelemişlerdir. Sonuçlar max-min stratejisinin operatörlerin yetenek seviyelerini daha çok geliştirdiğini göstermektedir. Azadeh ve diğ. (2014) bulanık veri zarflama analizi (BVZA) ile bulanık benzetim tekniklerini kombine ederek çok ürünlü hücreli üretim sistemlerinde çalışan tahsisi problemini ele almışlardır. BVZA birçok seviyede belirsizlik içeren benzetim sonuçlarını değerlendirmek için kullanılmıştır. Egilmez ve diğ. (2014) hem operasyon sürelerinin hem de talebin belirsiz olduğu hücreli üretim ortamında stokastik yetenek tabanlı çalışan tahsis problemi ile ilgilenmişlerdir. Dört aşamalı hiyerarşik bir methodoloji içerisinde 3 farklı stokastik doğrusal olmayan optimizasyon modeli geliştirmişlerdir. Önerilen methodoloji gerçek üretim ortamından alınan bir örnek üzerinde uygulanmıştır. Niakan ve diğ. (2016) çalışan atama problemini çevresel ve sosyal kriterler ile birlikte ele almak için iki amaçlı bir optimizasyon modeli önermişlerdir. Problemin çözümünde kullanmak üzere NSGA-II-MOSA melez algoritmasını geliştirmişlerdir. Feng ve diğ. (2017) hücre oluşturma ve çalışan atama problemlerini beraber ele alan doğrusal bir optimizasyon modeli geliştirmişlerdir. Modelde üretim planlama, ürün rotalama, parti bölme, işyükü dengeleme ve çalışanların hücrelere atanma kararlarını eş zamanlı olarak ele almışlardır. Problemin NP-Zor olması nedeniyle parçacık sürü optimizasyonu (PSO) ve doğrusal programlamayı birleştiren bir yaklaşım önererek ilk aşamada PSO ve ikinci aşamada CPLEX optimizasyon programını kullanmışlardır. Çalışmanın sonuçlarında çalışanların hücrelere belirlenen sınırlardan fazla atanmasının hem çalışan maliyetleri hem de kullanım oranları açısından faydalı olduğunu belirtmişlerdir. Kuo ve Liu (2017) çalışmalarında hücre için çalışan transferini ele alarak iki aşamalı bir methodoloji ile toplam çalışan sayısının en küçüklenmesini amaçlamışlardır. İlk aşamada tamsayılı programlama kullanılarak görevlerin istasyonlara atanma konfigürasyonları belirlenmiş ve ikinci aşamada karma tamsayılı programlama modelinde ilk aşamanın sonuçları kullanılarak toplam çalışan sayısının en küçüklenmesi amaçlanmıştır. Gerçek bir üretim ortamından elde edilen verilerle

birlikte gerçekleřtirmiř oldukları deneyler sonucunda önerdikleri yöntembilimin çalışan etkinliğini arttırdığını belirtmiřlerdir.

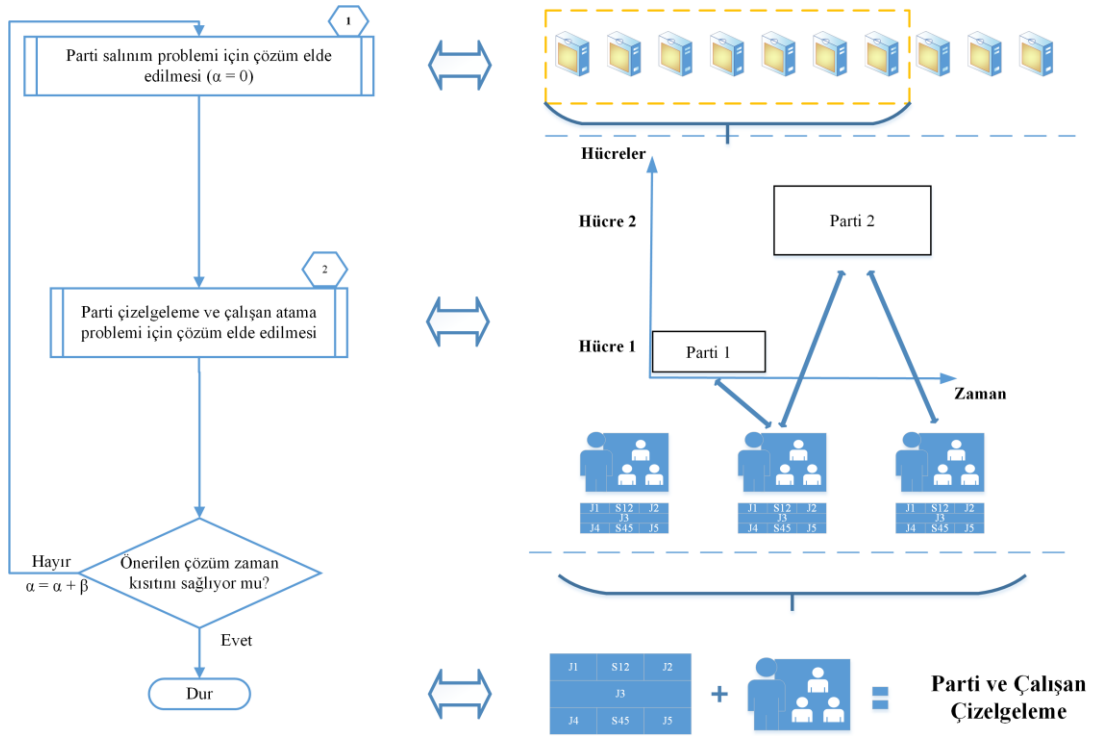
HÜS’ de çalışan tahsisi/ataması ile ilgili çalışmalar incelendiğinde çalışanların gerek parti işleme gerekse de parça işleme durumları için hücrelere statik olarak atandıkları görölmektedir. Bir önceki kısımda açıklandığı üzere parti çizelgeleme problemi ile birlikte çalışanların da eş zamanlı dinamik olarak çizelgelenmesi problemi literatürde daha önce incelenmemiřtir. Atama/tahsis çalışmalarında incelendiğinde ařağıdaki sonuçlar çıkarılmaktadır.

- Hücrelere çalışan ataması/tahsisi problemi genellikle hücrelerin oluşturulması ve yüklenmesi problemleri ile birlikte ele alınmıřtır. Her iki problemin beraber ele alınmasının nedeni ÇKK sistemlerle ilgilenmek olduğı belirtilmiřtir.
- Çalışanların etkin gruplar oluşturması ile ilgili literatürde çalışmalar bulunmaktadır. Ancak bu çalışmaların sayısı yeterli değıldir.
- Çalışanların hem hücre içi hem de hücreler arası hareketlerini en azlamak amacıyla birçok çalışma gerçekleřtirilmiř ve bu çalışmalarla çalışan hareketleri minimize edilerek çalışma oranları arttırılmaya çalışılmıřtır.
- Literatürde çalışanlarla ilgili farklı senaryolar hem optimizasyon modelleri hem de benzetim çalışmaları ile birlikte analiz edilmiřtir. Çalışanların sistem performansı üzerindeki etkisinin en çok çalışan yoğun hücresel üretim ortamlarında gözlemlendiğı belirtilmiřtir.
- Tek amaçlı optimizasyon modellerinin yanısıra birden çok amacı eş zamanlı olarak ele alan çalışmalarda bulunmaktadır.

Bu tez kapsamında melez üretim hücrelerinde parti çizelgeleme problemi çalışanların partilere bağımlı ve dinamik olarak çizelgelenmesi problemi ile beraber ele alınmaktadır. Bu sayede hem parti çizelgeleme problemi hem de çalışanların atanması problemi dinamik bir şekilde ele alınmıř olmaktadır. Hücrelerin oluşturulması ve yüklenmesi problemi ele alınmamakla birlikte bu bilgiler birer parametre olarak kullanılmıřtır. Bu sayede hücresel üretim sistemlerinde PÇÇA problemi ile ilgili olarak literatürde ki boşluk doldurulmuř olacaktır.

3. ÖNERİLEN YÖNTEMBİLİM

Bu tez kapsamında HÜS’ de parti salınım (PS), parti çizelgeleme ve çalışan atama (PÇÇA) problemlerine bütünsel bir çözüm bulabilmek amacıyla iki aşamalı sıralı bir yöntembilim önerilmektedir. Önerilen yöntembilim ilk aşamasında parti salınım problemi için geliştirilen optimizasyon modeline, ikinci aşamasında parti ve çalışan çizelgeleme problemi için geliştirilen optimizasyon modeline yer verilmektedir.



Şekil 3.1 : Önerilen yöntembilim.

Önerilen yöntembilimin akış şeması ve uygulama benzetimine Şekil 3.1’ de yer verilmektedir.

Gelen siparişler üzerinden hangi partinin sisteme salınacağını belirlemenin parti salınım için geliştirilmiş olan matematiksel modelin en iyi sonucunun elde edilmesiyle bulunmaktadır. Parti salınım problemi için polinomsal süreler içerisinde en iyi çözümler bulunabildiği için sezgisel yöntemin içerisinde yer verilmemektedir. Sisteme salınım yapılan partiler belirlendikten sonra parti ve çalışan çizelgeleme

matematiksel modeli ile küçük boyutlu problemler (makine başına parti sayısı < 40) için en iyi çözüm bulunmaktadır. Büyük boyutlu problemler için önerilen meta-sezgisel algoritma kullanılmaktadır. İkinci aşamanın sonucunda elde edilen çözüm çizelgeleme periyodu kısıtını sağlıyorsa uygun bir çözüm olarak kullanılmaktadır. Eğer zaman kısıtını sağlamıyorsa bu durumda sisteme daha az sayıda parti salınımı gerçekleştirilerek önerilen yöntem bilim tekrar uygulanmaktadır.

3.1 Parti Salınım Problemi

Yayın taraması kısmında belirtildiği üzere, sipariş salınımı (SS) üretim kontrol politikası işyükü kontrolünden bir önceki adım olabileceği gibi işyükü kontrolünün ana adımlarından bir tanesi de olabilir. Bu tez kapsamında sipariş salınımı yerine parti salınımı (PS) terimi kullanılmaktadır.

Parti salınım politikası iki ana amaçla gerçekleştirilmektedir (Thürer ve diğ, 2015). Bu amaçlardan ilki, iş istasyonlarında dengesiz işyüklemesinin önlenmesi iken ikincisi süreç içi stokun dengelenmesidir. Temel anlamda iki farklı salınım politikası bulunmaktadır; periyodik salınım ve sürekli salınım (Fernandes ve diğ, 2016). Periyodik salınım işlerin üretim ortamına salınmadan önce bir ön bekleme havuzunda sıralanması ve periyodik zaman aralıklarında salınmasını gerektirir (Land, 2006). Literatürdeki yayınlar incelendiğinde işlerin ön bekleme havuzunda sıralanmasının genellikle ihmal edildiği ve hangi işlerin salınacağı kararına öncelik verildiği görülmektedir (Thürer ve diğ. 2014). Partilerin salınımı gerçekleştirilirken bir üst limit belirlenerek (bu tez kapsamında vardiya süresi) işlerin teslim zamanlarına önem verilerek üretime alınmaları sağlanır. Bu sayede hem iş istasyonlarının işyükü dengesini sağlayan hem de süreç içi stoku azaltan parti karışımı üretim ortamına periyodik zaman aralıklarında salınmak üzere belirlenmiş olmaktadır. Ancak bu yöntemin iki bakımdan sakıncası bulunmaktadır. Belirlenen parti karışımı periyodik zaman aralıklarında salındığı için herhangi bir darboğaz kaynağın (iş istasyonu) önündeki besleyici stok bittiğinde hemen tepki verilemediği için beklenmedik bir boş kalma durumu oluşmaktadır. Ayrıca üretim sistemine salınacak ürünlerin karışımı belirlenirken istasyonların işyükleri arasındaki dengenin sağlanması adına herhangi bir iş istasyonu (kısıtlı bir iş istasyonu olmayabilir) diğer iş istasyonunun işyükü durumundan dolayı atıl durumda kalabilir (Kanet, 1988; Land ve Gaalman, 1998). Diğer taraftan, sürekli salınım politikası bir üst limit yerine

yeniden yenileme noktası yöntemini alt limit tetikleyicisi olarak kullanmaktadır. Örneğin bir darboğaz kaynağın önündeki besleyici stok belirli bir seviyeye indiğinde o kaynağın ihtiyaç duyduğu ürünler ön bekleme havuzundan tetiklenerek üretim ortamına salınmaktadır. Bu sayede sürekli olarak salınım politikası işletilebilmektedir. Sürekli salınım politikası, periyodik salınım politikasının doğurmuş olduğu olumsuzlukları önleyebilmektedir ve bu anlamda üstün olduğu söylenebilir (Sabuncuoglu ve Karapınar, 1999). Ancak iş istasyonları arasındaki işyükü dengesi için bir üst sınır kullanılmadığı için işyükü dengesizliklerine neden olabilmektedir (Qi ve diğ, 2009). Bu durumun önüne geçebilmek adına literatürde gerçekleştirilen çalışmalarda melez sürekli salınım politikaları önerilmektedir (Thürer ve diğ, 2014; Fernandes ve Carmo-Silva, 2011). Ayrıca SS kontrol politikasına alternatif olarak Kanban ve CONWIP (Constant workin process) üretim kontrol sistemleri de kullanılabilmektedir (Thürer ve diğ, 2014). Ancak bu sistemler kullanılarak verilen salınım kararlarında işyükü dengesizlikleri göz önünde bulundurulmamaktadır (Germs ve Riezebos, 2010).

Bu tez kapsamında ele alınan parti salınım problemi için geliştirilen ve periyodik salınıma olanak sağlayan ikili tamsayılı programlama modeline aşağıda yer verilmektedir.

İndisler

i : Partiler için indis ($i=1, \dots, N$)

k : Hücreler için indis ($k=1, \dots, K$)

S_k : Partilerin kümeleri ($k=1, \dots, K$)

Parametreler

α : Çevrim süresi değişim oranı

q_i : i . parti içerisindeki parça sayısı

ca_k : k . hücrenin kapasitesi

Değişkenler

cn_i : i . parti için minimum hücre çevrim süresi

cx_i : i . parti için maximum hücre çevrim süresi

cy_i : i . parti için hücre çevrim süresi

FT_i : i . parti içerisindeki ilk parçanın temin süresi

p_i : i . partinin işlenme süresi

Karar değişkenleri

re_i : Eğer i . parti sisteme salındıysa, 1; aksi halde, 0

Matematiksel model

Amaç fonksiyonu

$$Max \text{ amaç} = \sum_{i=1}^N re_i * q_i \quad (3.1)$$

Kısıtlar

$$\sum_{i \in S_k} p_i \times re_i \leq ca_k \quad k=1,...,K \quad (3.2)$$

$$p_i = cy_i \times (q_i - 1) + FT_i \quad i=1,...,N \quad (3.3)$$

$$cy_i = cn_i + (cx_i - cn_i) * \alpha \quad i=1,...,N \quad (3.4)$$

$$re_i \in \{0,1\} \quad (3.5)$$

Denklem 3.1 amaç fonksiyonunu göstermektedir. Amaç maksimum sayıda parçanın üretim sistemine salınmasını sağlamaktır. Denklem 3.2 hücrelerin kapasite aşımalarını önlemek üzere oluşturulmuştur. Kapasiteler süre cinsinden birer parametre olarak belirlenmektedir. Denklem 3.3 her parti için işlenme süresinin hesabında kullanılmaktadır. Denklem 3.4 hücre çevrim süresinin hesabında kullanılmaktadır. Bu süre hesaplanırken çalışan atamalarına yer verilmemektedir ve hücre içerisindeki toplam operasyon süresi olarak değerlendirilmektedir. Denklem 3.4 hücre çevrim süresinin hesabında kullanılmaktadır. Bu denklem ile çevrim süresi hesaplanırken çevrim süresi değişim oranı (α) kullanılmaktadır. Bu oran sıfıra eşit olduğunda hücre kapasitesi maksimum düzeyde kullanılmış olurken, bire eşit olduğunda minimum seviyede kullanılmış olmaktadır. Kapasite maksimum seviyede kullanıldığında sisteme salınan parça sayısı yüksek olurken, minimum seviyede kullanıldığında salınan parça sayısı düşük olmaktadır. Önerilen yöntemimizin son kısmında eğer elde edilen çizelge zaman sınırını aşıyorsa çevrim süresi değişim oranının artırılması ve sisteme daha az parça salınımı yapılarak zaman sınırı aşımının önlenmesi amaç edinilmiştir.

Bu tez kapsamında geliştirilen optimizasyon modeliyle periyodik salınım gerçekleştirilmektedir. Hücrelerin birbirinden bağımsız olması aralarındaki işyükü dengesizliklerini önemsiz kılmaktadır, çünkü hücrelerden çıkan parçalar bitmiş ürün olarak değerlendirilmektedir. Bu nedenle amaç dengesizliklerin ortadan kaldırılması olarak değil maksimum sayıda parçanın sisteme salınması olarak belirlenmiştir. Optimizasyon modelindeki amaç maksimum sayıda parçanın sisteme salınması

olduğu için ön bekleme havuzunda bekleyen partiler içerdikleri parça sayısına göre büyükten küçüğe sıralanmakta ve eğer kapasite kısıtı sağlanıyorsa sırayla sisteme salınmaktadırlar. Böylece parti büyüklüğü bir öncelik kuralı olarak değerlendirilmiş olmaktadır. Geliştirilmiş olan optimizasyon modeli polinomsal zamanda çözülebildiği için en iyi çözümler GAMS® 23.5/CPLEX 12.2 paket programı ile elde edilmiştir.

3.2 Parti Çizelgeleme Problemi

Yayın taraması kısmında da belirtildiği üzere HÜS‘ de parti çizelgeleme problemini çalışan atama problemi ile birleştirerek çalışanların partilere bağımlı olarak çizelgelenmesine olanak sağlayan bir çalışma bulunmamaktadır.

Ele alınan parti çizelgeleme probleminde hücre çevrim sürelerinin çalışan atanmalarına bağlı olarak değiştiği gerçeği göz önünde bulundurulmaktadır. Literatürde hücre çizelgeleme çalışmaları tek ve çok hücreden oluşan HÜS‘ler için incelenmektedir. Her iki tip HÜS yapısı için gerçekleştirilen çizelgeleme çalışmalarında makine bazında ilk olarak ürün aileleri sırasının belirlenmesi ve daha sonra her ürün ailesindeki parçalar için makine bazında sıraların belirlenmesi gerçekleştirilmektedir. Ancak bu tez kapsamında grup çizelgeleme probleminden bağımsız olarak yetenekli çalışanların hücre çevrim süreleri üzerine olan etkilerini ele alan, çalışanların partilere bağımlı olarak çizelgelenmesini sağlayan ve hücre içerisinde tek parça akışın uygulandığı melez akış tipi üretim hücrelerinde parti çizelgeleme problemine yer verilmektedir. Bu nedenle problem hücre çizelgeleme problemi yerine parti çizelgeleme problemi olarak adlandırılmaktadır.

Bu tez kapsamında, parti çizelgeleme problemi çalışan atama problemi ile birleştirilerek birden çok bağımsız melez üretim hücresinden oluşan hücresel üretim sistemi (HÜS) için ele alınmaktadır. Bu problem için çalışanların farklı yeteneklere sahip oldukları göz önünde bulundurulmuş ve çalışan havuzları oluşturulmuştur.

Melez üretim hücrelerinden oluşan HÜS bir çeşit hücresel üretim sistemidir ve birbirinden bağımsız ve paralel birden çok hücreden oluşmaktadır. Yalnızca otomatik veya montaj (manuel) operasyonlarının gerçekleştiği HÜS‘ nin aksine hem otomatik hem de manuel operasyonlar gerçekleştirilmektedir ve bu nedenle hücreler melez olarak adlandırılmaktadır (Hyer ve Wemmerlov, 2002; Yılmaz ve Durmuşoğlu,

2017). Yetenekli çalışan atama problemi özellikle melez üretim hücreleri için önemlidir. Manuel operasyonların olduğu hücrelerde çalışanların yetenek seviyeleri hücre çevrim süresinin belirlenmesinde kullanılmaktadır ve dolayısıyla hücre çıktı sayısını etkilemektedir (Yılmaz ve diğ., 2016; Hyer ve Wemmerlov, 2002).

Çift kaynak kısıtlı (ÇKK) üretim sistemlerinde kapasiteden kaynaklı çıktı sayılarının değişimi hem çalışan sayılarına hem de makine sayılarına bağlıdır (Hottenstein ve Bowman, 1998). Özellikle manuel operasyonların gerçekleştirildiği sistemlerde çalışan sayısının iş istasyonu sayısından az olması çalışanları önemli bir kaynak haline getirmektedir. Bu çeşit bir hücresel üretim sistemi ÇKK HÜS olarak adlandırılmaktadır ve bilimsel literatürde ÇKK üretim sistemleri sıklıkla ele alınmaktadır (Costa ve diğ., 2013). ÇKK HÜS' de operatörler genellikle kısıtlayıcı kaynaklar olmaktadır ve hem hücre içi hem de hücreler arası hareketler gerçekleştirmektedirler (Xu ve diğ., 2011; Satoğlu ve Suresh 2009). Çalışanların çevrim sürelerini ve dolayısıyla parti işlem sürelerini etkiledikleri melez üretim hücrelerinden oluşan bir sistem ÇKK HÜS için verilebilecek başlıca örneklerden birtanesidir (Hyer ve Wemmerlov, 2002; Yılmaz ve diğ., 2016). Çalışanların hücrelere atanmasının yeteneklerine bağlı olduğu ve çalışan yetenek seviyelerinin hücre çevrim süresini etkilediği bu sistemlerde hücre çevrim süreleri ciddi oranlarda değişim gösterebilmekte ve bu durum özellikle kısıtlı çalışan sayısına sahip ÇKK HÜS' de PÇÇA problemlerini çok daha önemli hale getirmektedir.

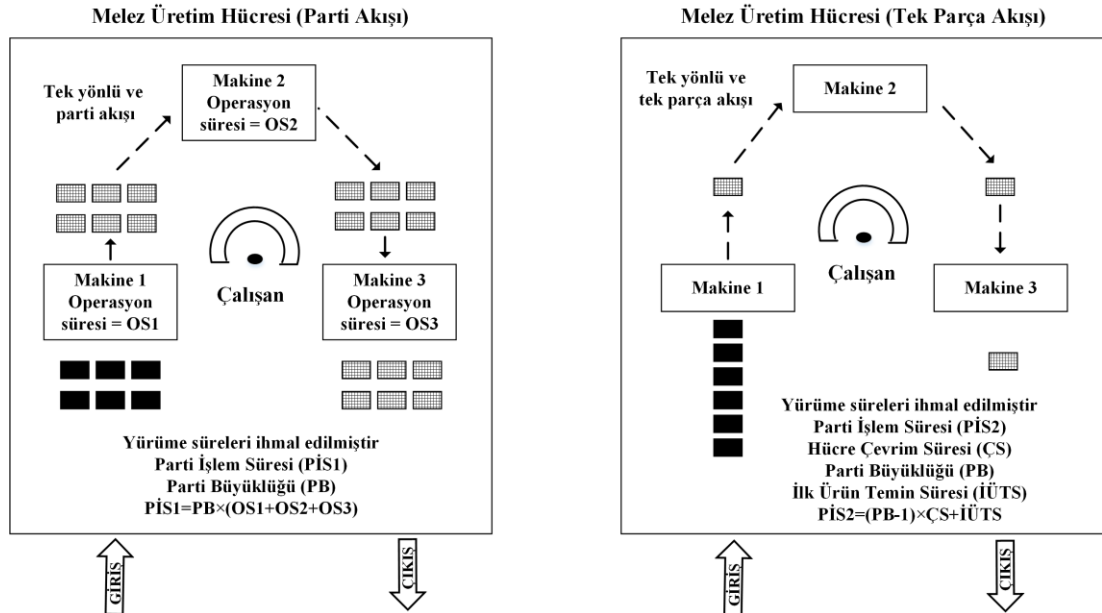
3.2.1 Problemin tanımlanması

Bu tez kapsamında ÇKK HÜS' de PÇÇA problemi ele alınmaktadır. Bu sayede partilerle birlikte çalışanlarında çizelgelenmesi sağlanmaktadır. Karar verilmesi gereken problemlerler şunlardır: partilerin işlenmeye başlanma zamanlarının belirlenmesi ve çalışanların partilere bağımlı olarak hücrelerdeki operasyonlara atanması.

K tane birbirinden bağımsız çalışan yoğun akış tipi melez üretim hücrelerinden oluşan bir ÇKK HÜS bulunmaktadır. Her üretim hücresi M tane makineden oluşmaktadır ve ürün ailelerine tahsis edilmişlerdir. Hücrelerin yapısı ve hangi partinin hangi üretim hücresinde işleneceği önceden bilinmektedir. Bütün partiler sıfır zamanında işlenmeye hazır olarak üretim öncesi ön bekleme havuzunda beklemektedirler. Partilerin salınım öncelikleri parti büyüklüğüne göre

belirlenmektedir. Her hücreye birtane ürün ailesi atandığı için partiler arası hazırlık süreleri ihmal edilmektedir. Bir partinin hücrede işlenmesine başladıktan sonra işlem yarıda kesilmemekte ve başka bir partinin hücrede işlenmesine izin verilmemektedir. Her hücrede uygulanan tek parça ve tek yönlü akış sayesinde partilerin içerisindeki parçalar hücre içerisindeki makinelerde sırayla işlenmektedirler. Tek yönlü akış hücre içerisinde parçaların geriye doğru hareket etmediği anlamına gelmektedir. Hücre içerisindeki her makinede hem manuel hem de otomatik operasyonlar gerçekleştirilebilmektedir ve her makine bir operasyona denk gelmektedir. Tüm makinelerde gerçekleştirilen operasyonlar ise ana operasyon olarak adlandırılmaktadır. Eğer bir parti için bir kayıp (atlanan) operasyon (KO) söz konusu ise parti içerisindeki parçalar ilgili operasyonu (makineyi) atlayarak işlenmeye bir sonraki makineden devam ederek ana operasyonlarını tamamlamaktadırlar. Her çalışan bir yetenek seviyesine sahiptir ve hücredeki operasyonları bu yetenek seviyesi üzerinden gerçekleştirerek parti işleme süresini etkilemektedir.

Melez üretim hücrelerinde parça akışlarının sağlanması için uygulanabilecek iki ana yöntem bulunmaktadır. Bu yöntemlerden ilki tek parça akışı olarak diğeri parti akışı olarak adlandırılmaktadır. Her iki akış tipi Şekil 3.2' de gösterilmektedir.



Şekil 3.2 : Melez üretim hücrelerinde uygulanabilecek akış tipleri.

Tek parça akışının uygulanması ile birlikte hücre içerisinde makinelerde işlenmeye devam eden parçalar parti içerisindeki parçaların tamamlanmasını beklemek zorunda

değillerdir (Satoğlu ve diğ, 2010). Tek parça akış uygulandığında parti işlem süresinin hesap edilmesinde hücre çevrim süresi (ÇS) kullanılabilir. Bu durumda parti işlem süresi, parti büyüklüğünün bir eksiği (PB-1) ile ÇS çarpıldıktan sonra ilk ürün temin süresiyle (İÜTS) toplanarak bulunur $((PB-1) \times \text{ÇS} + \text{İÜTS})$. Diğer taraftan, parti akışının uygulandığı hücrelerde tüm parti bitmeden bir diğer makinedeki operasyona herhangi bir parçanın geçişine izin verilmemektedir ve tüm parçaların işlenmesinin beklenmesi gerekmektedir. Bu durumda, parti işlem sürelerinin hesabında hücre çevrim süresi kullanılamamaktadır (Yılmaz ve Durmuşoğlu, 2017). Parti akışı uygulandığında parti işlem süresi operasyon sürelerinin toplamıyla $(OS1+OS2+OS3)$ parti büyüklüğünün (PB) çarpılması sonucu bulunur. Ek olarak her iki akış tipini yansıtan Şekil 3.2' de örnekler için yürüme süreleri ihmal edilmiştir.

Yukarıda yer verilen açıklamalardan da anlaşılacağı üzere melez üretim hücresi içerisinde tek parça akışın uygulanması parti akışının uygulanmasına nazaran parti işlem sürelerini büyük oranlarda düşürebilmektedir. Ek olarak tek parça akışı uygulandığında elde edilen parti işlem süresi ile parti akışı uygulandığında elde edilen parti işlem süresinin birbirlerine eşit olmadığı problemin yapısal özellikleri kısmında açıklanmıştır (Lütfen kısım 3.2.4.1 Lemma 2 inceleyiniz).

Bir parti bir parça olarak değerlendirildiğinde, tek parça akışını uygulamak mümkün değildir. Ele alınan problem için herhangi bir parti bir parça olarak değerlendirilmemekte ve hücreler içerisinde tek parça akışı uygulanmaktadır. Eğer problem parça veya iş çizelgeleme problemi olarak adlandırılırsa bu durumda her parti birer parça olarak değerlendirilecek ve hücre çevrim süresi kullanılmadığından dolayı parti işlem sürelerinde büyük oranda artışlar olacaktır. Bu durum problemin neden parti çizelgeleme problemi olarak adlandırıldığının sebeplerinden bir tanesidir. Diğer sebep ise partilerin makineler bazında değil hücreler bazında çizelgelenmesidir. Yani, parti sıraları makineler için değil hücreler için belirlenmektedir. Eğer partiler makineler bazında çizelgelenirse bu durum problemin hücre çizelgeleme problemine indirgenmesine yol açar ki hücre çizelgeleme problemi daha öncede açıklandığı üzere ayrı bir problem türüdür ve bu tez kapsamında yer verilmemektedir. Bu tezin ana amacı bütünsel bir yöntem bilim içerisinde hem parti salınım hem de parti çizelgeleme problemlerini yetenekli çalışan

atama ve çizelgeleme problemi ile birlikte tek parça akışın uygulandığı bir ÇKK HÜS için ele almaktır.

Yukarıda yer verilen hücre çevrim süresi bir partiye ait iki tamamlanmış parça arasında geçen süre olarak tanımlanmaktadır. Hücrede gerçekleştirilen ana operasyona birden fazla çalışan atanabileceği için çevrim süresinin hesabında çalışanlar tarafından sağlanan çevrim süreleri karşılaştırılmaktadır (denklemler 3.13 ve 3.14). Ayrıca hücre çevrim süresi hücre içerisindeki maksimum operasyon süresinden küçük olamayacağı gibi (denklem 3.19), tüm manuel operasyon ve yürüme sürelerinin toplamından da büyük olamaz (Hyer ve Wemmerlov, 2002; Yılmaz ve Durmuşoğlu, 2017; Black ve Hunter, 2003; Miltenburg, 2001).

PÇÇA problemi için çalışanların farklı yetenek seviyelerine sahip oldukları göz önünde bulundurulmaktadır. Costa ve diğ. (2014) çalışmalarında çalışanlar için üç farklı yetenek seviyesi belirlemişlerdir. Bu yetenek seviyeleri yüksek, orta ve düşük olarak belirlenmiştir. Belirledikleri bu seviyeler aynı zamanda bu tez kapsamında da kullanılmaktadır. Yetenek seviyesi düşük bir çalışan yetenek seviyesi orta ve yetenek seviyesi yüksek bir çalışandan daha az uzman iken, yetenek seviyesi orta bir çalışan yetenek seviyesi düşük bir çalışandan daha uzman ancak yetenek seviyesi yüksek bir çalışandan daha az uzmandır. Çalışanların uzmanlık düzeyleri gerçekleştirdikleri manuel operasyonların sürelerini etkilemektedir. Örneğin manuel bir operasyon yetenek seviyesi yüksek bir çalışan tarafından gerçekleştirildiğinde yetenek seviyesi orta bir çalışan tarafından gerçekleştirilmesine nazaran daha kısa sürede tamamlanmaktadır. Bu durum hücre çevrim sürelerinin çalışanların yetenek seviyeleri tarafından etkilenmesine neden olmaktadır. Benzer şekilde, hücre çevrim süreleri hücrelere atanan çalışan sayılarından da etkilenmektedir. Dolayısıyla, parti işlem süreleri hem çalışanların yetenek seviyelerinden hem de hücrelere atanan çalışan sayılarından etkilenmektedir. Bu nedenle, PÇÇA probleminde parti başlangıç zamanları ile hangi çalışanın hangi hücredeki hangi makineye hangi parti için atanacağı kararları eş zamanlı olarak verilmelidir.

Literatürde çalışan kaynağının değerlendirilmesi üç şekilde karşımıza çıkmaktadır: yenilenebilir, yenilenemez ve çift kısıtlı kaynaklar (Edis ve diğ, 2013). Bir çalışan kaynağı birden fazla çakışmayan operasyona atanabiliyorsa yenilenebilir bir kaynak olarak adlandırılmaktadır aksi halde yenilenemez kaynak olarak adlandırılmaktadır. Diğer taraftan, eğer bir kaynak çizelgeleme periyodu boyunca hem yenilenebilir hem

de yenilenemez karakteristiklere sahipse bu durumda çift kısıtlı kaynak olarak adlandırılmaktadır. Çalışmalarda çalışanların genellikle yenilenebilir veya yenilenemez olduğu varsayımı ile hareket edilmekte ve ele alınan problem bu durum göz önünde bulundurularak sınıflandırılmaktadır. Çalışan kaynağı göz önünde bulundurulduğunda iki çeşit çizelgeleme probleminden bahsedilebilir: dinamik çizelgeleme ve statik çizelgeleme problemleri (Yılmaz ve Durmuşoğlu, 2017). Dinamik çizelgeleme problemleri yenilenebilir çalışan kaynağı atanması ile ilgilenirken, statik çizelgeleme problemleri yenilenemez kaynak atanması ile ilgilenmektedir. Bu tez kapsamında, çalışan yetenek seviyelerinin toplam çalışan maliyeti üzerindeki etkisini daha net görebilmek adına çalışanların yenilenebilir olarak atandığı gerçeğinden hareket edilerek parti çizelgeleme problemi dinamik bir çerçevede ele alınmaktadır.

3.2.2 Optimizasyon modeli

Tez kapsamında ele alınan PÇÇA problemi için geliştirilmiş tamsayılı programlama optimizasyon modeline bu kısımda yer verilerek detaylı olarak açıklaması yapılmıştır. Ele alınan problem belirli varsayımlar altında dinamik paralel makineler esnek kaynak çizelgeleme (PMEKÇ) problemine indirgenmektedir (Daniels ve diğ., 1996; Edis ve diğ., 2013). Bu problem NP-Zor sınıfında olup çözümü için sezgisel veya meta-sezgisel yöntemler kullanılan bir problemidir. Ele aldığımız problemin NP-Zor yapısına kısım 3.2.3' de detaylı olarak yer verilmektedir.

Problem için tanımlanmış olan indisler, parametreler, değişkenler, karar değişkenleri ve optimizasyon modeline aşağıda yer verilmektedir.

İndisler

i, j : Partiler için indisler ($i, j = 1, \dots, N$)

f, g : Parçalar için indisler ($f, g = 1, \dots, F$)

z : Çalışanlar için indis ($z = 1, \dots, Z$)

m : Makineler için indis ($m = 1, \dots, M$)

k, l : Hücreler için indisler ($k, l = 1, \dots, K$)

t : Zaman periyotları için indis ($t = 1, \dots, T$)

S_k : Partilerin kümeleri ($k = 1, \dots, K$)

Parametreler

N_k : k . hücreye atanan toplam parti sayısı

N : Toplam parti sayısı, $\sum_{k=1}^K N_k = N$

F : Toplam parça sayısı, $\sum_{i=1}^N q_i = F$

q_i : i . parti içerisindeki toplam parça sayısı

$e_{z,i,m}$: i . partideki parçaların/ürünlerin m . makinede z . çalışan için manuel operasyon süreleri

$r_{i,m}$: i . partideki parçaların/ürünlerin m . makinede otomatik operasyon süreleri

o_i : i . parti için çalışanların toplam yürüme süreleri

a_i : i . parti için çalışanların ortalama yürüme süreleri

$u_{i,m}$: Eğer m . makine i . partinin operasyonlarında kullanılıyorsa, 1; aksi halde, 0

W_i : i . partinin ana operasyonuna atanabilecek maksimum çalışan sayısı

ζ_z : z . çalışanın birim zaman maliyeti

T : Çizelgeleme periyodundaki zaman periyodu sayısı (T , yayılma süresi için üst sınır değeri)

Değişkenler

C : Yayılma süresi

C_k : k . hücre için yayılma süresi

c_i : i . partinin tamamlanma zamanı

p_i : i . partinin işlenme süresi

cy_i : i . parti için hücre çevrim süresi

$wI_{z,i}$: Eğer z . çalışan i . partinin ana operasyonuna atandıysa, 1; aksi halde, 0

$n_{i,j}$: Eğer i . partinin operasyon başlama zamanı j . partininkinden küçükse, 1; aksi halde, 0

$nI_{i,j}$: Eğer i . partinin tamamlanma zamanı j . partininkinden büyükse, 1; aksi halde 0

cn_i : i . parti için minimum hücre çevrim süresi

cx_i : i . parti için maksimum hücre çevrim süresi

$s_{i,k}$: i . partinin başlangıç zamanı

$v_{z,i,m}$: i . partideki parçaların m . makinede z . çalışan için operasyon süreleri

d_i : i . partideki parçalar için en yüksek operasyon süresi

FT_i : i . partideki ilk parçanın temin süresi

Karar değişkenleri

$x_{i,t}$: Eğer i . partinin operasyonları t . zamanında bitirildiyse, 1; aksi halde, 0

$b_{i,j}$: Eğer i . parti j . partiden sonra işleniyorsa, 1; aksi halde, 0 ($i, j \in S_k$)

$w_{z,i,m}$: Eğer z . çalışan i . parti için m . makineye atandıysa, 1; aksi halde, 0

Problem için geliştirilmiş olan matematiksel model aşağıda yer almaktadır.

Amaç fonksiyonu

$$\text{Min } \text{amaç} = C \times \sum_{z=1}^Z \xi_z \quad (3.6)$$

Kısıtlar

$$C \geq c_i \quad i = 1, \dots, N \quad (3.7)$$

$$c_i = \sum_{t=1}^T t \times x_{i,t} \quad i = 1, \dots, N, t = 1, \dots, T \quad (3.8)$$

$$\sum_{t=1}^T x_{i,t} = 1 \quad i = 1, \dots, N \quad (3.9)$$

$$c_i \geq (c_j + p_i) - T \times (1 - b_{i,j}) \quad i \in S_k, j \in S_k / k = 1, \dots, K \quad (3.10)$$

$$b_{i,j} + b_{j,i} = 1 \quad i \in S_k, j \in S_k / k = 1, \dots, K \quad (3.11)$$

$$p_i = cy_i \times (q_i - 1) + \sum_{z=1}^Z \sum_{m=1}^M w_{z,i,m} \times (e_{z,i,m} + r_{i,m}) + o_i \quad i = 1, \dots, N \quad (3.12)$$

$$w_{z,i,m} \times e_{z,i,m} + r_{i,m} \leq cy_i \quad i = 1, \dots, N, z = 1, \dots, Z, m = 1, \dots, M \quad (3.13)$$

$$\sum_{m=1}^M w_{z,i,m} \times (a_i + e_{z,i,m}) \leq cy_i \quad i = 1, \dots, N, z = 1, \dots, Z \quad (3.14)$$

$$\sum_{z=1}^Z w_{z,i,m} = u_{i,m} \quad i = 1, \dots, N, m = 1, \dots, M \quad (3.15)$$

$$w1_{z,i} \geq w_{z,i,m} \quad i = 1, \dots, N, z = 1, \dots, Z, m = 1, \dots, M \quad (3.16)$$

$$\sum_{z=1}^Z w1_{z,i} \leq W_i \quad i = 1, \dots, N \quad (3.17)$$

$$w1_{z,i} + w1_{z,j} \leq 3 - (n_{i,j} + n1_{i,j}) \quad i \in S_k, j \in S_l / k, l = 1, \dots, K, k \neq l \quad (3.18)$$

$$(c_j - p_j) - (c_i - p_i) \leq T \times n_{i,j} \quad i \in S_k, j \in S_l / k, l = 1, \dots, K, k \neq l \quad (3.19)$$

$$c_i - (c_j - p_j) \leq T \times n1_{i,j} \quad i \in S_k, j \in S_l / k, l = 1, \dots, K, k \neq l \quad (3.20)$$

$$x_{i,t}, w_{z,i,m}, w1_{z,i}, n_{i,j}, n1_{i,j}, b_{i,j}, x_{i,t} \in \{0,1\}; cy_i, p_i, c_i \geq 0 \quad (3.21)$$

Denklem 3.6 amaç fonksiyonunu ifade etmektedir. Amaç fonksiyonunun ilk kısmı yayılma süresini (C) ifade ederken ikinci kısmı çalışan havuzunun birim zaman maliyetini ifade etmektedir. Yayılma süresi ile birim zaman maliyeti çarpıldığında toplam çalışan maliyeti elde edilmektedir. Bu sayede amaç fonksiyonu toplam çalışan maliyeti ve yayılma sürelerini eş zamanlı olarak enazlamak üzere en uygun

çalışan havuzunun seçilmesini sağlamaktadır. Denklem 3.7 yayılma süresinin her partinin tamamlanma zamanından daha büyük olması gerektiğini ifade etmektedir. Denklemler 3.8 ve 3.9 her partinin tamamlanma zamanının bulunmasında kullanılmaktadır. Denklem içerisinde kullanılan T parametresi yayılma süresi için bir üst sınır ifade etmektedir ve her parti T süresine kadar bir zaman periyodunda tamamlanmak zorundadır. T değerinin hesaplanmasında kısım 3.4.2.2 içerisinde yer verilen teorem 2 kullanılmaktadır. Denklem 3.10 sayesinde herhangi bir hücrede aynı anda yalnızca bir partinin işlenmesine izin verilmektedir. Denklem 3.11 her hücre içerisindeki partilerin sırasının bulunmasında kullanılmaktadır (ya i . parti $i \in S_k$ j . partiden $j \in S_k$, önce işlenecek ya da tam tersi). Denklem 3.12 atanan çalışanlarla birlikte değişim gösteren parti işlem sürelerinin hesaplanmasında kullanılmaktadır. Tek parça akışı sayesinde parti işlem süreleri hesaplanırken hücre çevrim süresi kullanılabilir. Denklem 3.12'nin ikinci kısmı i . parti içerisindeki ilk parçanın temin süresinin hesabında kullanılmaktadır. Denklemler 3.13 ve 3.14 her parti için hücre çevrim sürelerinin hesaplanmasında kullanılmaktadır. Bu denklemlerin oluşturulmasında Miltenburg (2001) çalışmasından yararlanılmıştır. Bir melez üretim hücresinde, bir parti için hücre çevrim süresi yürüme süreleri ve manuel operasyon sürelerinin toplanması ile bulunmaktadır. Ayrıca hücre çevrim süresi minimum hücre çevrim süresinden ve her çalışan için hesaplanan hücre çevrim sürelerinden büyük ya da eşit olmalıdır. Denklem 3.15 çalışanların operasyonları gerçekleştirmek üzere operasyonlara (makinelere) atanmalarını garanti etmektedir. Denklem 3.16 bir çalışan bir operasyona (makineye) atandıysa aynı zamanda o operasyonu içeren ana operasyona da (hücreye de) atanmasını sağlamaktadır. Hücre içerisindeki her makine bir operasyona karşılık gelirken, hücre bir makine olarak değerlendirildiğinde ana operasyona karşılık gelmektedir. Hücrenin içerisindeki tüm operasyonların aynı anda değerlendirilerek ana operasyon olarak adlandırılmasının nedeni aynı anda bir çalışanın birden fazla ana operasyona yani hücreye atanmasının önüne geçebilmektir. Denklem 3.17 bir ana operasyona atanan toplam çalışan sayısının o ana operasyona atanabilecek maksimum çalışan sayısından küçük veya eşit olması gerektiğini ifade etmektedir. Çalışanlar yenilebilir kaynaklar olarak değerlendirildiğinden çizelgeleme periyodu boyunca birden fazla operasyona atanabilirler. Ancak aynı anda farklı hücrelerdeki ana operasyonlara atanamazlar. Bu denklem sayesinde modelin çözümünde daha kısa sürede en iyi çözümlere ulaşılabilmektedir. Denklemler 3.18, 3.19 ve 3.20 çalışan uygunluk kısıtlarını göz

önünde bulundurmaktadırlar. Denklem 3.18 herhangi bir çalışanın aynı anda en fazla bir ana operasyona atanabileceğini ifade ederken aynı anda gerçekleşen operasyonlara atanmamasını garanti etmektedir. Denklemler 3.19 ve 3.20 çalışan ana operasyonları belirlemek için kullanılmaktadırlar. Bu kısıtlar vasıtasıyla çalışanların çizelgeleme periyodu boyunca dinamik olarak operasyonlara atanması sağlanmaktadır. Denklem 3.21 ikili tamsayı ve işaret kısıtlamalarını ifade etmektedir. Matematiksel modelde kullanılan cx_i ve cn_i parametrelerinin hesabında kullanılan denklemlere aşağıda yer verilmektedir (Yılmaz ve diğ, 2016).

$$cx_i = \max \left(\left(\sum_{m=1}^M \max_{\forall z} (e_{z,i,m}) + o_i \right); d_i \right) \quad i=1,...,N \quad (3.22)$$

$$cn_i = d_i \quad i=1,...,N \quad (3.23)$$

Denklem 3.22 maksimum hücre çevrim süresinin hesabında kullanılmaktadır. Bu değer hesaplanırken, hücredeki tüm makinelerin olası en yüksek manuel operasyon sürelerinin toplamı ile toplam yürüme süresi toplanarak hücredeki en yüksek operasyon süresi ile karşılatırılmaktadır.

Denklem 3.23 minimum hücre çevrim süresinin hesabında kullanılmaktadır. Bu değer aynı zamanda hücredeki en yüksek operasyon süresine eşittir.

Hücredeki en yüksek operasyon süresinin hesaplanmasında denklemler 3.24 ve 3.25 kullanılmaktadır.

$$d_i = \max_{\forall m,z} (v_{z,i,m}) \quad i=1,...,N \quad (3.24)$$

$$v_{z,i,m} = e_{z,i,m} + r_{i,m} \quad i=1,...,N, m=1,...,M, z=1,...,Z \quad (3.25)$$

Hücredeki en yüksek operasyon süresi o hücredeki tüm makinelerin en yüksek operasyon süresine eşittir. Ayrıca hücredeki makinelerin operasyon süreleri manuel ve otomatik operasyon sürelerinin toplamı ile hesap edilmektedir.

Hücreye atanabilecek maksimum çalışan sayısının hesabında denklem 3.26 kullanılmaktadır.

$$W_i = \left[cx_i / cn_i \right]^+ \quad i=1,...,N \quad (3.26)$$

Bir hücreye atanabilecek maksimum çalışan sayısı o hücre için elde edilen maksimum hücre çevrim süresinin minimum hücre çevrim süresine bölünmesi ile hesap edilmektedir. Bu değeri hesaplanmasının nedeni hücreye gereksiz çalışan

atamasının önüne geçmektir. Bu sayede hem matematiksel modelin hem de sezgisel algoritmaların çözüm süresinde azalmalar olmaktadır.

Matematiksel modeldeki denklemlerde kullanılan ve yayılma süresi için üst sınır değerine eşit olan T denklem 3.27 ile hesap edilmektedir.

$$T = \sum_{k=1}^K \sum_{i \in S_k} (cx_i \times (q_i - 1) + FT_i) \quad (3.27)$$

T üst sınır değeri tüm parti işlem sürelerinin toplamı ile hesap edilmektedir. Parti işlem sürelerinin hesabında ise maksimum hücre çevrim süreleri kullanılmaktadır.

Denklem 3.28 partilerin işlenme başlangıç zamanlarının hesabında kullanılmaktadır. Bu denklemde p_i parti işlem süresini ifade etmektedir ve tamamlanma zamanından çıkarıldığında başlangıç zamanı bulunmaktadır.

$$s_i = c_i - p_i \quad i = 1, \dots, N \quad (3.28)$$

Matematiksel modeldeki denklem 3.12' nin ikinci kısmında, partinin ilk parça için temin süresinin hesabı yer almaktadır. İlk parça temin süresi denklem 3.29 kullanılarak hesap edilmektedir.

$$FT_i = \sum_{z=1}^Z \sum_{m=1}^M w_{z,i,m} \times (e_{z,i,m} + r_{i,m}) + o_i \quad i = 1, \dots, N \quad (3.29)$$

İlk parça tüm makinelerden (operasyonlardan) geçtiğinden ve önünde bir parça olmadığından dolayı tüm operasyon süreleriyle yürüme süresinin toplanarak temin süresinin hesap edilmesi gerekmektedir. Parti işlem süresi hesabında (denklem 3.12) ise ilk parçanın temin süresi ile ilk parça çıkarıldıktan sonra geriye kalan parça sayısının çevrim süresi ile çarpımı sonucu elde edilen değer toplamı kullanılmaktadır.

3.2.3 Problemin NP-Zor yapısı

Ele alınan PÇÇA problemi için aşağıdaki varsayımlarda bulunulduğunda problem statik PMEKC problemine (Edis ve diğ., 2013) indirgenmektedir.

- Çalışanlar yenilenemez ve birbirine eş kaynaklardır.
- Tüm hücreler birbiri ile aynı özellikleri taşımaktadır.
- Her parti bir parça olarak kabul edilmektedir.

Literatürde statik PMEKC probleminin çözümü için Daniels ve diğ. (1996) tarafından problemin polinomsal zamanda çözülmesi için algoritma geliştirilmiştir.

Yukarıdaki diğer varsayımları değiştirmeden ilk varsayım aşağıdaki varsayımla değiştirildiğinde, problem dinamik PMEKC problemine indirgenmektedir.

- Çalışanlar yenilenebilir ve birbirine eş kaynaklardır.

Dinamik PMEKC probleminin NP-Zor bir problem olduğu akademik literatürde kanıtlanmıştır (Edis ve diğ, 2013; Daniels ve diğ, 1996; Kellerer ve Strusevich 2008).

Benzer şekilde ele alınan problem aşağıdaki varsayımlar ile değerlendirildiğinde, belirlenen makineler ile kaynak kısıtlı paralel makineler çizelgeleme (KKPMC) problemine indirgenmektedir. Bu varsayımda bulunulduğunda parti çizelgeleme problemi çalışan çizelgeleme problemi ile birlikte ele alınmaktadır.

- Çalışanlar yenilenebilir ve birbirine eş olmayan kaynaklardır.
- Operasyon işlem süreleri çalışan atamalarından etkilenmemektedir.

Akademik literatürde belirlenen makineler ile kaynak kısıtlı paralel makineler çizelgeleme (KKPMC) problemi NP-Zor sınıfında bir problem olarak kabul edilmektedir (Edis ve diğ, 2013). Ayrıca KKPMC probleminde operasyon işlem sürelerinin sabit olduğu varsayılmaktadır. (İlgilenen okuyucular paralel makineler çizelgeleme problemleri için farklı varsayımlar altında gerçekleştirilen sınıflandırmalara Edis ve diğ. (2013) çalışmasından ulaşabilir)

Bu tez kapsamında PCCA problemi tanıtılmakta ve toplam çalışan maliyetinin en küçülenmesi amaçlanmaktadır. Problem kapsamında eş olmayan hücreler, yenilenebilir ve eş olmayan çalışan kaynağı, sabit olmayan işlem süreleri ve çalışan çizelgeleme yer aldığı için problemin NP-Zor sınıfında olduğu açıkça anlaşılmaktadır. Yukarıda da açıklandığı üzere belirli varsayımlar gerçekleştirildiğinde dahi problem NP-Zor sınıfındaki problemlere indirgenmektedir.

Problem NP-Zor sınıfında bir problem olduğundan dolayı problem büyüklüğünün artması ile birlikte problemin çözümü için harcanan zaman üssel olarak artmaktadır. Problem için oluşturulan küçük boyutlu örneklerin en iyi çözümlerine ulaşmak için harcanan sürelere kısım 5 içerisinde yer verilmektedir. Bu nedenle problemin çözümünde kullanılmak üzere sezgisel yöntemler önerilmektedir.

3.2.4 Problemin yapısal özellikleri, alt ve üst sınırlar

Ele alınan problemin özelliklerinin daha net açıklanabilmesi adına aşağıda problemin yapısal özelliklerine yer verilerek belirli varsayımlar altında elde edilen çözümlerin değişimi incelenmekte ve bazı ispatlar yapılmaktadır.

3.2.4.1 Yapısal özellikler

Lemma (Önerme) 1. *Problem için belirlenen bir çizelge $\Pi = (b_1, b_2, \dots, b_N)$ bulunmaktadır ve bu çizelgeden elde edilen çözüm bu durumlarda değişmektedir: (1) aynı hücrede işlenen iki partinin pozisyonu (sırası) birbirleri ile değiştirildiğinde, (2) aynı hücrede işlenen iki partinin içerisindeki parçaların pozisyonları birbirleri ile değiştirildiğinde. Bu çizelgeden elde edilen çözüm bu durumda değişmemektedir: (3) bir partinin içerisindeki iki parçanın pozisyonları birbiri ile değiştirildiğinde.*

İspat

1. Genelliği kaybetmeden, b_i ve b_j partilerinin aynı hücrede işlendiğini ele alalım $b_i, b_j \in S_k$ ($k=1, \dots, K$; $K \geq 1$). Aynı hücre içerisinde b_i partisi b_j partisinden önce işlendiğinde, tamamlanma zamanları $c_i^1 = s_i^1 + p_i^1$, $c_j^1 = s_j^1 + p_j^1$ ve parti işleme süreleri $p_i^1 = cy_i \times (q_i - 1) + FT_i$, $p_j^1 = cy_j \times (q_j - 1) + FT_j$ için yer verilen eşitlikler elde edilmektedir. Her iki partinin pozisyonları birbirleri ile değiştirildiğinde, güncellenen parti işlem süreleri $p_i^2 = cy_i \times (q_i - 1) + FT_i$, $p_j^2 = cy_j \times (q_j - 1) + FT_j$ için yer verilen eşitlikler elde edilmektedir. Bu eşitliklerden parti işleme sürelerinin (p_i, p_j) pozisyon değişiminden etkilenmediği anlaşılmaktadır. Ancak hem parti başlangıç zamanları (s_i, s_j) hem de parti tamamlanma zamanları (c_i, c_j) , partilerin sırasına ve çalışanların uygunluklarına göre değişim göstermektedirler. Güncellenen parti tamamlanma zamanları $c_i^2 = s_i^2 + p_i^2$, $c_j^2 = s_j^2 + p_j^2$ için yer verilen eşitlikler elde edilmektedir. Bu durumda, yayılma süresi C^1 ve güncellenen yayılma süresi C^2 arasındaki farkın b_i ve b_j partilerinin tamamlanma zamanları ile güncellenen tamamlanma zamanları arasındaki farka bağımlı olduğu anlaşılmaktadır. Böylece $c_i^1 - c_i^2 \neq 0$ ve $c_j^1 - c_j^2 \neq 0$ eşitsizlikleri elde edilmektedir. Birden çok hücrenin olduğu durum ele alındığında ($k=K$; $K > 1$), b_i ve b_j partilerinin başlangıç zamanları (s_i, s_j) ve tamamlanma zamanları (c_i, c_j) değişmektedirler.

ve bu deęişim dięer partilerin tamamlanma sürelerini etkilemektedir. Bu sayede, çözümün hem tek hücreli hem de çok hücreli sistemlerde deęişim gösterdiği sonucuna varılmaktadır.

2. Genellięi kaybetmeden, b_i ve b_j partilerinin aynı hücrede işlendiğini ele alalım $b_i, b_j \in S_k, J_f \in b_i, J_g \in b_j$ ($f, g=1, \dots, F; k=1, \dots, K; K \geq 1$). Aynı hücre içerisinde b_i partisi b_j partisi beraber işlendiğinde, tamamlanma zamanları $c_i^1 = s_i^1 + p_i^1$, $c_j^1 = s_j^1 + p_j^1$ ve parti işleme süreleri $p_i^1 = cy_i \times (q_i - 1) + FT_i$, $p_j^1 = cy_j \times (q_j - 1) + FT_j$ için yer verilen eşitlikler elde edilmektedir. J_f ve J_g parçalarının pozisyonları birbirleri ile deęiştirildiğinde, güncellenen parti işlem süreleri $p_i^2 = cy_i \times (q_i - 1) + FT_i$, $p_j^2 = cy_j \times (q_j - 1) + FT_j$ için yer verilen eşitlikler elde edilmektedir. Bu eşitliklerde kullanılan hücre çevrim süreleri cy_i ve cy_j farklı partilerdeki parçalara bağımlıdır (denklemler 3.13 ve 3.14). Manuel operasyon süreleri aynı olmadığı için parçaların pozisyonları deęiştirildiğinde parti işlem süreleri deęişim göstermektedir. Bu durumda, yayılma süresi C^1 ve güncellenen yayılma süresi C^2 arasındaki farkın b_i ve b_j partilerinin tamamlanma zamanları ile güncellenen tamamlanma zamanları arasındaki farka bağımlı olduğu anlaşılmaktadır. Böylece $c_i^1 - c_i^2 \neq 0$ ve $c_j^1 - c_j^2 \neq 0$ eşitsizlikleri elde edilmektedir. Birden çok hücrenin olduğu durum ele alındığında ($k=K; K \geq 1$), b_i ve b_j partilerinin başlangıç zamanları (s_i, s_j) ve tamamlanma zamanları (c_i, c_j) deęişmektedirler ve bu deęişim dięer partilerin tamamlanma sürelerini etkilemektedir. Bu sayede, çözümün hem tek hücreli hem de çok hücreli sistemlerde deęişim gösterdiği sonucuna varılmaktadır.
3. Genellięi kaybetmeden, b_i partisinin bir hücrede işlendiğini ele alalım $b_i \in S_k, J_f \in b_i, J_g \in b_i$ ($k=1, \dots, K; K \geq 1$). Aynı parti içerisinde J_f paçası J_g parçasından önce işlendiğinde tamamlanma zamanı $c_i^1 = s_i^1 + p_i^1$ ve parti işleme süresi $p_i^1 = cy_i \times (q_i - 1) + FT_i$ için yer verilen eşitlikler elde edilmektedir. Bu eşitliklerde kullanılan hücre çevrim süresi cy_i aynı parti içerisindeki parçaların işleme sırasından bağımsızdır (denklemler 3.13 ve 3.14). Bu durumda, yayılma süresi C^1 ve güncellenen yayılma süresi C^2 arasındaki farkın b_i

partisinin tamamlanma zamanı ile güncellenen tamamlanma zamanı arasındaki farka bağımlı olduğu anlaşılmaktadır. Böylece $C^1 - C^2 = 0$ ve eşitliği elde edilmektedir. Birden çok hücrenin olduğu durum ele alındığında ($k=K$; $K>1$), b_i partisinin işlem süresi p_i değişmemektedir ve diğer partilerin başlangıç ve tamamlanma zamanlarını etkilememektedir. Bu sayede, çözümün hem tek hücreli hem de çok hücreli sistemlerde değişim gösterdiği sonucuna varılmaktadır.

Lemma (Önerme) 2. Parti b_i bir parça olarak değerlendirildiğinde parti işlem süresi p_i değişmektedir.

İspat Genelliği kaybetmeden, b_i partisinin bir hücrede işlendiğini ele alalım $b_i \in S_k$ ($k=1, \dots, K$; $K=1$). Parti b_i bir parça olarak değerlendirildiğinde hücre içerisinde tek parça akışını uygulamak mümkün olmamaktadır ve bu durumda parti işlem süresi için $p_i^1 = \left(\sum_{z=1}^Z \sum_{m=1}^M w_{z,i,m} \times (e_{z,i,m} + r_{i,m}) + o_i \right) \times q_i$, $\sum_{z=1}^Z \sum_{m=1}^M w_{z,i,m} \times (e_{z,i,m} + r_{i,m}) + o_i = FT_i$ eşitlikleri elde edilmektedir. Parti b_i işlenmesi sırasında hücre içerisinde tek parça akış uygulandığında parti işlem süresi için $p_i^2 = cy_i \times (q_i - 1) + FT_i$ eşitliği elde edilmektedir (denklem 3.12). Eğer parti büyüklüğü q_i bire eşitse, parti işlem süresi için $p_i^1 = \sum_{z=1}^Z \sum_{m=1}^M w_{z,i,m} \times (e_{z,i,m} + r_{i,m}) + o_i = FT_i$, $p_i^2 = FT_i$ eşitlikleri elde edilirken, aksi halde, parti işlem süresi için $p_i^1 = FT_i \times q_i$, $p_i^2 = cy_i \times (q_i - 1) + FT_i$ eşitlikleri elde edilmektedir. Böylece parti büyüklüğü bire eşit olduğunda $p_i^1 = p_i^2$ eşitliği elde edilirken, parti büyüklüğü birden büyük olduğunda $p_i^1 \neq p_i^2$ eşitsizliği elde edilmektedir.

Lemma (Önerme) 3. k . hücre için en küçük yayılma süresi $C_k = \sum_{i \in S_k} p_i$.

İspat Genelliği kaybetmeden, problem için belirlenen bir çizelge $\Pi = (b_1, b_2, \dots, b_k, \dots, b_N)$ olduğunu varsayalım. Kolay bir şekilde $C_k = \sum_{i \in S_k} p_i = \sum_{i \in S_k} (cy_i \times (q_i - 1) + FT_i)$ eşitliğini elde edebiliriz. Bu sayede, b_i partisi için hücre çevrim süresi cy_i minimum hücre çevrim süresi cn_i eşit olduğunda k . hücre için en küçük yayılma süresi için $C_k = \sum_{i \in S_k} p_i = \sum_{i \in S_k} (cn_i \times (q_i - 1) + FT_i)$ eşitliği elde edilmektedir. Benzer şekilde, b_i partisi için hücre çevrim süresi cy_i maximum hücre çevrim süresi cx_i eşit olduğunda k .

hücre için en küçük yayılma süresi için $C_k = \sum_{i \in S_k} p_i = \sum_{i \in S_k} (cx_i \times (q_i - 1) + FT_i)$ eşitliği elde edilmektedir.

Lemma (Önerme) 4 Hücreler için en küçük yayılma sürelerinin toplamı

$$\sum_{k=1}^K C_k = \sum_{i=1}^N p_i.$$

İspat Genelliği kaybetmeden Lemma 3 temel alındığında,

$$\sum_{k=1}^K C_k = \sum_{k=1}^K \sum_{i \in S_k} p_i = \sum_{k=1}^K \sum_{i \in S_k} (cy_i \times (q_i - 1) + FT_i) \text{ eşitliği elde edilmektedir. Bu}$$

sayede, b_i partisi için hücre çevrim süresi cy_i minimum hücre çevrim süresi cn_i eşit olduğunda hücreler için en küçük yayılma sürelerinin toplamı için

$$\sum_{k=1}^K C_k = \sum_{k=1}^K \sum_{i \in S_k} p_i = \sum_{k=1}^K \sum_{i \in S_k} (cn_i \times (q_i - 1) + FT_i) \text{ eşitliği elde edilmektedir. Benzer}$$

şekilde, b_i partisi için hücre çevrim süresi cy_i maximum hücre çevrim süresi cx_i eşit olduğunda hücreler için en küçük yayılma sürelerinin toplamı için

$$\sum_{k=1}^K C_k = \sum_{k=1}^K \sum_{i \in S_k} p_i = \sum_{k=1}^K \sum_{i \in S_k} (cx_i \times (q_i - 1) + FT_i) \text{ eşitliği elde edilmektedir. Buradan}$$

hücreler için en küçük yayılma süreleri toplamının tüm partilerin işlem sürelerinin toplamına eşit olduğu sonucu çıkarılmaktadır.

3.2.4.2 Alt ve üst sınırlar

Teorem 1 Toplam çalışan maliyeti (amaç fonksiyonu) için alt sınır değeri

$$AS = \sum_{z=1}^Z \zeta_z \times \sum_{k=1}^K \sum_{i \in S_k} p_i / K \text{ denklemi ile elde edilmektedir. Bu denklemde parti } b_i (b_i \in S_k; k=1, \dots, K) \text{ işlem süresi için } p_i = (cn_i \times (q_i - 1) + FT_i) \text{ eşitliği kullanılmaktadır.}$$

İspat

Genelliği kaybetmeden elimizde alt sınır değerinden (AS) küçük AC değerine sahip

bir çözüm olduğunu varsayalım ($AC < \sum_{z=1}^Z \zeta_z \times \sum_{k=1}^K \sum_{i \in S_k} p_i / K$). Böylece,

$$AC \times K < \sum_{z=1}^Z \zeta_z \times \sum_{k=1}^K \sum_{i \in S_k} p_i \text{ eşitsizliği elde edilmektedir. Lemma 4 dikkate}$$

alındığında, parti b_i için minimum hücre çevrim süresi cn_i kullanıldığında

$$\sum_{k=1}^K \sum_{i \in S_k} p_i = \sum_{k=1}^K \sum_{i \in S_k} (cn_i \times (q_i - 1) + FT_i) \text{ elde edilmektedir. Bu nedenle}$$

$$AC < \sum_{z=1}^Z \zeta_z \times \sum_{k=1}^K \sum_{i \in S_k} p_i / K \text{ ve } AC \times K < \sum_{z=1}^Z \zeta_z \times \sum_{k=1}^K \sum_{i \in S_k} p_i \text{ eşitsizliklerinin Lemma 3}$$

ve Lemma 4 ile çelişkili olduğu açıkça anlaşılmaktadır. Sonuç olarak, alt sınır

denklemini için $AS = \sum_{z=1}^Z C_z \times \sum_{k=1}^K \sum_{i \in S_k} p_i / K$ ($p_i = (cn_i \times (q_i - 1) + FT_i)$) eşitliği kullanılmaktadır.

Teorem 2 Yayılma süresi için üst sınır değeri $\ddot{U}S = \sum_{k=1}^K \sum_{i \in S_k} p_i$ denklemini ile elde edilmektedir. Bu denklemde parti b_i ($b_i \in S_k$; $k=1, \dots, K$) işlem süresi için $p_i = (cx_i \times (q_i - 1) + FT_i)$ eşitliği kullanılmaktadır.

İspat

Genelliği kaybetmeden elimizde üst sınır değerinden ($\ddot{U}S$) büyük AC değerinde bir yayılma süresi olduğunu varsayalım ($\ddot{U}S > \sum_{k=1}^K \sum_{i \in S_k} p_i$). Tüm hücrelerde çalışan sayısının bire eşit olması durumunda çalışan uygunluğunu tüm çizelgeleme periyodu için sağlamak adına her hücrenin çalışanı sırasıyla kullanması gerekmektedir. Lemma 4 dikkate alındığında, hücrelerdeki minimum yayılma sürelerinin toplamının $\sum_{k=1}^K C_k = \sum_{i=1}^N p_i$ eşitliği ile elde edildiği bilinmektedir. Parti b_i için maksimum hücre çevrim süresi cx_i kullanıldığında $\sum_{k=1}^K \sum_{i \in S_k} p_i = \sum_{k=1}^K \sum_{i \in S_k} (cx_i \times (q_i - 1) + FT_i)$ eşitliği elde edilmektedir. Bu nedenle $\ddot{U}S > \sum_{k=1}^K \sum_{i \in S_k} p_i = \sum_{k=1}^K \sum_{i \in S_k} (cx_i \times (q_i - 1) + FT_i)$ eşitsizliğinin Lemma 3 ve Lemma 4 ile çelişkili olduğu açıkça anlaşılmaktadır. Sonuç olarak, yayılma süresi için üst sınır denklemini olarak $\ddot{U}S = \sum_{k=1}^K \sum_{i \in S_k} p_i = \sum_{k=1}^K \sum_{i \in S_k} (cx_i \times (q_i - 1) + FT_i)$ kullanılmaktadır.

4. META-SEZGİSEL ALGORİTMALAR

Kısım 3.2.3 de belirtildiği üzere ele alınan PÇÇA problemi NP-Zor sınıfında bir problemidir. Bu nedenle problemin çözümünde kullanılmak üzere genetik algoritma (GA) tabanlı beş farklı algoritma geliştirilmiş ve birbirleriyle karşılaştırmaları sonucu problemin çözümünde kullanılmak üzere bir algoritma seçilmiştir. Her algoritmada farklı öncelik kuralı kullanılarak hangi öncelik kuralının daha etkin olduğu belirlenmeye çalışılmıştır.

4.1 PÇÇA Probleminin Kodlanması ve Örnek Bir Uygulama

Algoritmaların problem için uyarlanması için bir kodlama yöntemi geliştirilmelidir. Bu kodlama yöntemi probleme göre belirli özellikler içerebilmektedir ve algoritmanın performansını etkilemektedir (Park ve Kim, 1998).

Bu tez kapsamında, parti listesi (PL) kodlama yöntemi kullanılmıştır. Bu yöntemin temelinde aktivite listesi (AL) kodlama yöntemi bulunmaktadır (Kolisch, 1996). PL temel olarak partilerin çizelgelenme önceliğini esas almaktadır. Çizelgeleme öncelikleri için geliştirilmiş olan öncelik kuralları PL kodlama yönteminde uygulanmaktadır. Bir çözüm dört farklı satırdan oluşmaktadır. İlk satır partilerin pozisyonlarına, ikinci satır partilerin numaralarına, üçüncü satır partilere atanan çalışanlara ve son olarak dördüncü satır partilere atanan çalışanların hangi makinelere atandığına yer vermektedir. PL içerisindeki son iki satır ikinci satıra bağımlı olarak oluşturulmaktadır.

Problem için örnek uygulama verilerine Çizelge 4.1’ de yer verilmektedir. Bu veriler dört farklı partinin iki farklı birbirinden bağımsız melez üretim hücresinde üretildiğini göstermektedir. Parti 1 ve 2 ilk hücrede, parti 3 ve 4 ikinci hücrede üretilmektedirler. Parti i ($i=1 \dots 4$) için minimum ve maksimum hücre çevrim süreleri sırasıyla cn_i ve cx_i ile ifade edilmektedir. Bu değerlerin hesabında denklemler 3.22-3.25 kullanılmaktadır. Bu örnek problem için tüm çalışanların yetenek seviyeleri orta olarak kabul edilmiştir (kısım 5.1 çizelge 5.5).

Çizelge 4.1 : Örnek problem verileri.

	Çevrim sürei		Parti hacni	Mak. 1		Mak. 2		Mak. 3		Mak. 4		Yürüme
	Mak.	Min.		Ot.	Ma.	Ot.	Ma.	Ot.	Ma.	Ot.	Ma.	Toplam
Hücre 1	66	35	5	5	15	0	10	0	5	15	20	16
	86	30	10	0	20	0	15	5	5	0	30	16
Hücre 2	80	30	5	15	15	5	5	0	30	20	10	20
	55	20	10	10	10	0	5	0	20	20	0	20

Partilere atanabilecek maksimum çalışan sayısının belirlenmesinde denklem 3.26 kullanılmaktadır. Partilere atanacak çalışanların sayıları Çizelge 4.2’ de yer verilen değerler yardımıyla bulunmaktadır. Örneğin ikinci partiye atananan çalışan sayısı birden küçük üçten büyük olamaz. Eğer ikinci partinin işlenmesi için birinci hücreye dört tane çalışan atanırsa bu durumda atanmış olan fazladan bir çalışan olmaktadır ve bu fazla çalışanın hücreye atanmış olmasının çevrim süresi üzerine bir etkisi bulunmamaktadır.

Çizelge 4.2 : Maksimum ve minimum çalışan sayıları.

Çalışan sayıları			
		Maksimum	Minimum
Hücre 1	Parti 1	2	1
	Parti 2	3	1
Hücre 2	Parti 3	3	1
	Parti 4	3	1

Ele alınan problem için geliştirilen PL kodlama yöntemi Çizelge 4.3’ de gösterilmektedir. İlk satır partilerin çizelgelenme öncelikleri ile ilgilidir, ikinci satır parti numaralarını göstermektedir, üçüncü ve dördüncü satırlar ise çalışan uygunluk kısıtları (denklemler 3.18-3.20) ile ilgilidirler. Üçüncü satır çalışanların hücrelere atanma kararlarını gösterirken, dördüncü satır çalışanların atanmış oldukları hücrelerdeki makinelere atanma kararlarını göstermektedir. Üçüncü ve dördüncü satırlar çalışanların birbirleri ile çakışan operasyonlara atanmasını önlemektedirler. Önerilmiş olan PL kodlama yöntemi ile algoritmaların problem için uygulanabilmesi sağlanmaktadır ancak bu kodlama yönteminden çıkan çözümü bulabilmek için de bir

yöntem geliştirilmesi gerekmektedir. Ele alınan problem için seri çizelge şeması (SÇŞ) yöntemi uygulanmıştır. Bu yöntemin adımları bir sonraki başlık altında detaylı olarak açıklanmıştır.

Çizelge 4.3 : Parti listesi kodlama yöntemi.

Pozisyon	1		2		3	4	
Parti listesi	2		4		1	3	
Parti-çalışan	1	3	1	2	2	1	3
Çalışan-makine	1-2	3-4	2-3	1-4	1-2-3-4	3-4	1-2

Parti işlem sürelerinin hesaplanması için hücre çevrim süreleri cy_i ve ilk parça temin sürelerinin FT_i hesaplanması gerekmektedir. Hücre çevrim sürelerinin hesaplanmasında denklemler 3.13 ve 3.14 kullanılırken, ilk parça temin süreleri denklem 3.29 kullanılarak hesaplanmaktadır. Çizelge 4 hücre çevrim süreleri ile ilk parça temin sürelerini göstermektedir.

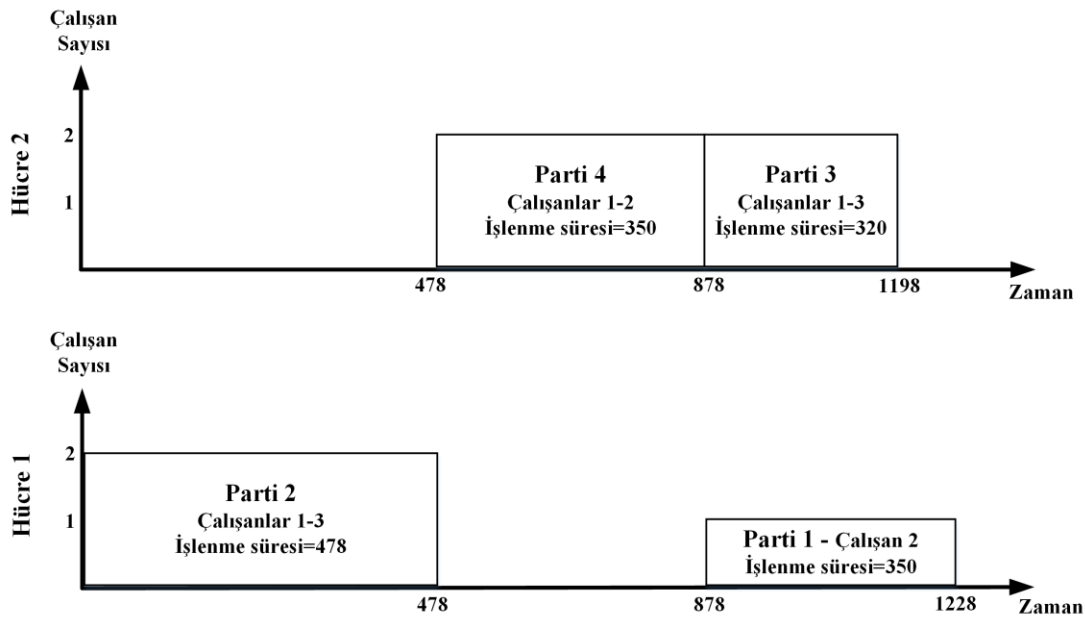
Çizelge 4.4 : Hücre çevrim süreleri ve ilk parça temin süreleri.

	Partiler	Çevrim Süresi	Çal. 1	Çal. 2	Çal. 3	Çevrim Süresi	İlk Parça Temin Süresi
Hücre 1	1	35	-	66	-	66	86
	2	30	43	43	-	43	91
Hücre 2	3	30	50	-	25	50	120
	4	20	35	20	-	35	85

4.1.1 PÇÇA probleminin çözümleme prosedürü ve amaç fonksiyonu

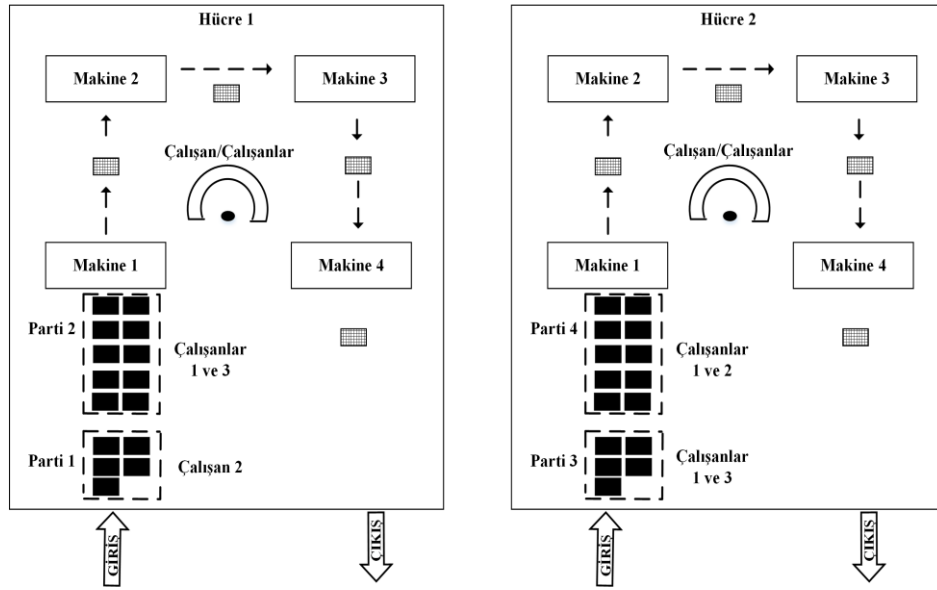
Genetik algoritmaların aşamalarında yerel en iyi çözümlere ulaşabilmek amacıyla populasyon içerisinde bulunan her çözümün amaç fonksiyonu değerinin bulunarak karşılaştırmalı olarak değerlendirilmesi gerekmektedir. Ele alınan PÇÇA problemi için geliştirilen matematiksel modelde amaç fonksiyonu olarak toplam çalışan maliyeti belirlenmiştir. Amaç fonksiyonu değerinin elde edilmesi için SÇŞ yöntemi her çözüme uygulanmıştır. Bu yöntem N ($i=1, \dots, N$) aşamadan ve çizelgelenmeyen partiler ve çizelgelenen partiler olarak adlandırılan iki farklı kümeden oluşmaktadır. Yöntemin ilk aşamasından önce tüm partiler çizelgelenmeyen partiler kümesinde yer

almaktadır ve her aşamada bu kümeden bir parti alınarak olurlu bir çizelge elde edecek şekilde çizelgelenmekte ve bu parti çizelgelenmeyen partiler kümesinden çıkarılarak çizelgelenen partiler kümesine eklenmektedir. Sistemde toplamda N adet parti bulunduğu dikkate alındığında N . aşama sonucunda çizelgelenmeyen partiler kümesinde hiçbir parti kalmayacak ve tüm partiler çizelgelenen partiler kümesinde olacaktır. Her aşamada hangi partinin çizelgelenmeyen partiler kümesinden alınacağı (öncelik) kararı Çizelge 4.3' ün birinci ve ikinci satırlarından çıkarılmaktadır. Parti çizelgelenirken çalışan uygunluk kısıtları dikkate alınarak partinin başlangıç süresi belirlenmektedir. Örneğin Çizelge 4.3 çözümü ele alındığında elde edilen çizelge Şekil 4.1' de görülmektedir. Bu çizelge sonucunda toplam maliyet 3684 olarak elde edilmektedir (Her çalışan için birim zaman maliyeti 1 olarak kabul edilmiştir).



Şekil 4.1 : Problemin çözülmesi ve örnek bir çizelge.

Şekil 4.1, Çizelge 4.3' de yer verilen örnek problem için elde edilen çizelgeyi göstermektedir. Şekil 4.2 ise örnek problem için sunulan HÜS yapısını göstermektedir. Toplamda dört farklı parti iki paralel birbirinden bağımsız melez üretim hücresinde işlenmektedir. Çalışan operasyonlar nedeniyle, çalışan atamaları direkt olarak hücre çevrim süresini ve dolaylı olarak toplam çalışan maliyetini etkilemektedir.



Şekil 4.2 : Örnek problem için melez hücre yapısı.

4.2 Genetik Algoritma ve Öncelik Kuralları

Genetik algoritma Holland (1992) tarafından geliştirilmiş popülasyon tabanlı metasezgisel bir algoritmadır. Geliştirildiği zamandan beri özellikle kombinatoriyel optimizasyon problemleri için uyarlanmış ve birçok problemde etkinliğini göstermiştir (Ünal ve diğ, 2012; Man ve diğ, 2012; Coley, 1999; Quagliarella ve diğ, 1998; Cortes ve diğ, 2001).

Genetik algoritmanın çalışma prensini üç ana aşamadan oluşmaktadır: seçim, çaprazlama ve mutasyon. Özellikle kesikli optimizasyon problemlerinde göstermiş olduğu etkinlik (Adewumi ve Ali 2010; Adewumi ve diğ, 2009; Naso ve diğ, 2007; Adamu ve Adewumi 2015) dikkate alınarak ele alınan PÇÇA problemi için uyarlanmasına karar verilmiştir.

Kısım 4.1’ de açıklanan problemin kodlanması ve çözümlenmesi prosedürleri genetik algoritma içerisinde kullanılmıştır. Genetik algoritmanın akış şemasına Şekil 4.3’ de yer verilmektedir.

Genetik algoritmanın ikinci aşamasında popülasyondaki her çözümün amaç fonksiyonu değeri hesaplanmaktadır. Bu amaç fonksiyon değeri hesaplanırken önerilen beş farklı öncelik kuralı kullanılmaktadır. Her öncelik kuralı genetik algoritma içerisinde ayrı olarak kullanıldığı için beş farklı algoritmanın varlığından bahsedilebilir. PÇÇA problemi için hangi öncelik

kuralının daha iyi sonuç verdiğini bulmak amacıyla algoritmalar birbirleri ile karşılaştırılmıştır. Aşağıda öncelik kuralları açıklanmaktadır.

Rassal Sıralama (RS): Tüm partilerin sırası rassal olarak belirlenmektedir. Örneğin Çizelge 4.3' ün ikinci satırı tamamen rassal olarak belirlenmektedir. RS öncelik kuralı genetik algoritma içerisinde kullanıldığında algoritma Genetik Algoritma Rassal Sıralama (GARS) olarak adlandırılmaktadır.

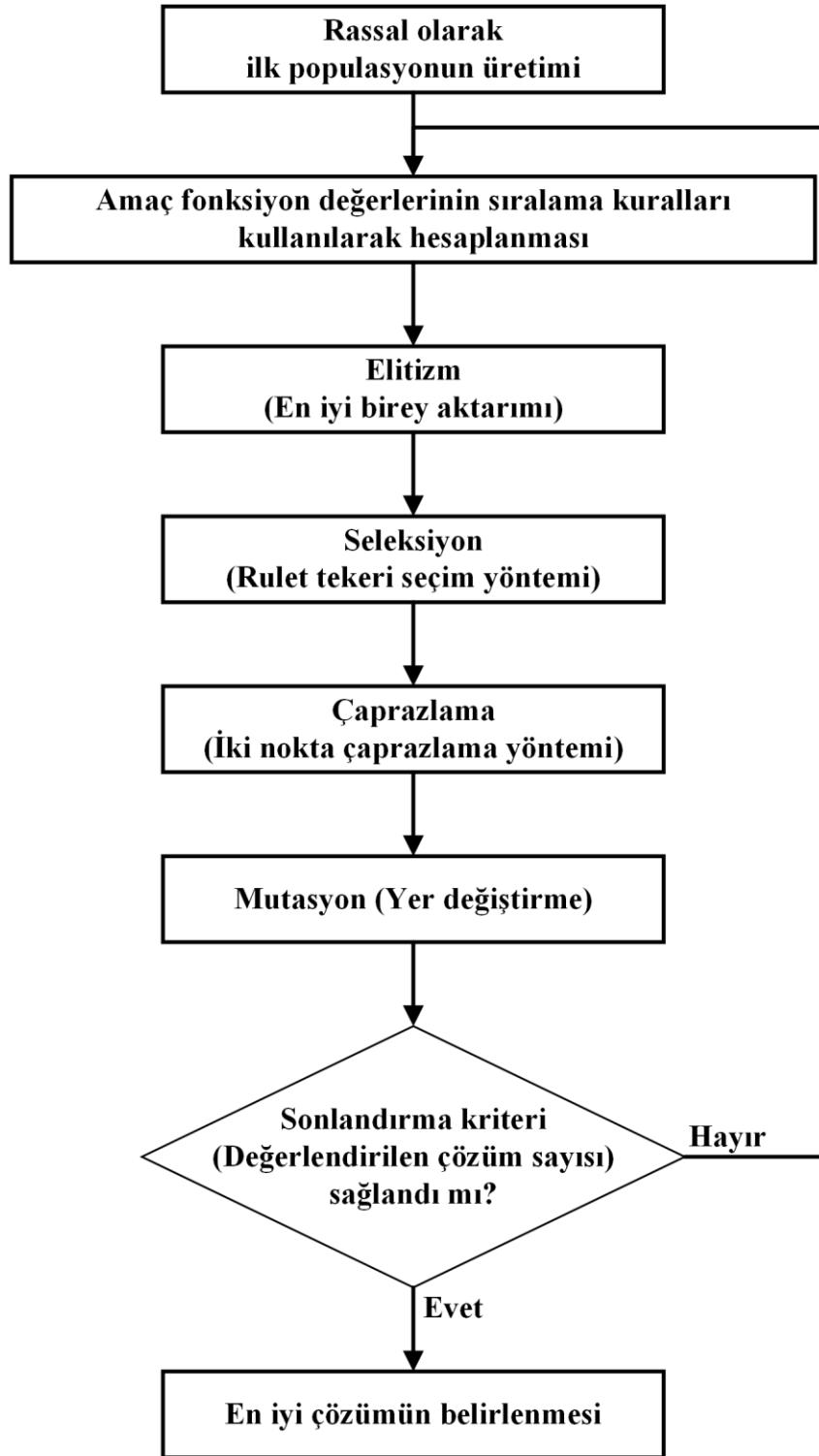
Kısa İşlenme Süresi Önce (KİSÖ): Tüm partiler işlenme sürelerine göre küçükten büyüğe sıralanmaktadır. Örneğin Çizelge 4.3' de tüm hücrelere ve makinelere atanan çalışanlar belli olduğu için tüm partilerin işlenme süreleri hesaplanarak küçükten büyüğe sıralanmaları sağlanmaktadır. KİSÖ öncelik kuralı genetik algoritma içerisinde kullanıldığında algoritma Genetik Algoritma Kısa İşlenme Süresi Önce (GAKİSÖ) olarak adlandırılmaktadır.

Uzun İşlenme Süresi Önce (UİSÖ): Tüm partiler işlenme sürelerine göre büyükten küçüğe sıralanmaktadır. Örneğin Çizelge 4.3' de tüm hücrelere ve makinelere atanan çalışanlar belli olduğu için tüm partilerin işlenme süreleri hesaplanarak büyükten küçüğe sıralanmaları sağlanmaktadır. UİSÖ öncelik kuralı genetik algoritma içerisinde kullanıldığında algoritma Genetik Algoritma Uzun İşlenme Süresi Önce (GAUİSÖ) olarak adlandırılmaktadır.

Hücre Bazlı Kısa İşlenme Süresi Önce (HBKİSÖ): Hücrelere atanan partiler işlenme sürelerine göre küçükten büyüğe sıralanmakta ve her seferinde bir hücreden yalnızca bir parti sıralanmak üzere seçilmektedir. Bu öncelik kuralının KİSÖ öncelik kuralından farkı aynı hücreden iki farklı partinin art arda sıralanmasının önüne geçilmesidir. Örneğin Çizelge 4.3' de tüm hücrelere ve makinelere atanan çalışanlar belli olduğu için tüm partilerin işlenme süreleri hesaplanarak hücre bazında küçükten büyüğe sıralanmaları sağlanmakta ve daha sonra her seferinde ayrı bir hücreden bir parti seçilerek sıralanmaktadır. HBKİSÖ öncelik kuralı genetik algoritma içerisinde kullanıldığında algoritma Genetik Algoritma Hücre Bazlı Kısa İşlenme Süresi Önce (GAHBKİSÖ) olarak adlandırılmaktadır.

Hücre Bazlı Uzun İşlenme Süresi Önce (HBUİSÖ): Hücrelere atanan partiler işlenme sürelerine göre büyükten küçüğe sıralanmakta ve her seferinde bir hücreden yalnızca bir parti sıralanmak üzere seçilmektedir. Bu öncelik kuralının UİSÖ öncelik kuralından farkı aynı hücreden iki farklı partinin art arda sıralanmasının önüne geçilmesidir. Örneğin Çizelge 4.3' de

tüm hücrelere ve makinelere atanan çalışanlar belli olduğu için tüm partilerin işlenme süreleri hesaplanarak hücre bazında büyükten küçüğe sıralanmaları sağlanmakta ve daha sonra her seferinde ayrı bir hücreden bir parti seçilerek sıralanmaktadır. HBUİSÖ öncelik kuralı genetik algoritma içerisinde kullanıldığında algoritma Genetik Algoritma Hücre Bazlı Uzun İşlenme Süresi Önce (GAHBUİSÖ) olarak adlandırılmaktadır.



Şekil 4.3 : Genetik Algoritma Akış Şeması.

Genetik algoritmada amaç fonksiyon değerlerinin hesaplanması için öncelik kurallarının uygulanmasından sonra en iyi amaç fonksiyon değerini veren iki birey bir sonraki popülasyona aktarılmak üzere ayırılmaktadır. Bu işlem elitizm olarak adlandırılmaktadır. Elitizm uygulandıktan sonra sırasıyla seleksiyon, çaprazlama ve mutasyon operatörleri uygulanmakta ve sonlandırma kriteri sağlandığında algoritma durdurularak en iyi sonucu veren birey çözüm olarak kullanılmaktadır.

4.2.1 Genetik algoritma operatörleri

Genetik algoritma içerisinde kullanılan üç temel operatör bulunmaktadır; seleksiyon, çaprazlama ve mutasyon (Li ve Gao, 2016).

Seleksiyon: GA algoritma uygulanırken her iterasyon sonunda bir sonraki aşamaya geçecek olan bireylerin belirlenmesi gerekmektedir. Bu bireyler belirlenirken uygunluk değerleri (amaç fonksiyonu) değerlendirilmektedir. Seçilen bireyler bir sonraki aşamadaki çocukları (bireyleri) oluşturacağı için ebeveyn olarak adlandırılırlar. Ebeveyn seçiminde her ne kadar amaç fonksiyon değerleri önemli olsada rassallık da söz konusudur. Bu nedenle literatürde birden fazla yöntem bulunmaktadır. Bu çalışma içerisinde rulet tekerleği seçimi (Miller ve Goldberg, 1995) yöntemi kullanılmaktadır. Bu yöntem uygulanırken tüm bireyler eş zamanlı olarak amaç fonksiyon değerlerine göre değişen olasılık değerleri alırlar. Bir rulet masası veya pasta olarak düşünürsek en iyi amaç fonksiyonu değerine sahip birey en büyük olasılık parçasını almaktadır. Daha sonra her seferinde iki ebeveyn belirleyecek şekilde rassal seçim gerçekleştirilir. Rassal seçim sonucunda yüksek olasılık dilimine sahip bireyin ebeveyn olma ihtimali daha yüksektir. Popülasyon büyüklüğüne eş sayıda ebeveyn seçildikten sonra ikili ebeveyn gruplarından iki farklı çocuk (yeni birey) meydana getirmek üzere çaprazlama operatörüne geçilmektedir.

Çaprazlama: Seleksiyon operatörü kullanılarak ebeveyn olarak seçilen birey çiftlerine bir sonraki popülasyonu oluşturmak üzere çaprazlama operatörü uygulanmaktadır. Bu tez kapsamında tek nokta çaprazlama operatörü kullanılmaktadır. Her ebeveyn çifti bir anne bir de baba olmak üzere *pcroos* olasılığı ile birlikte bir kız bir de erkek birey oluşturmak üzere

çaprazlanırlar. İlk olarak rassal bir t çaprazlama noktası belirlenir ve t noktasına kadar olan tüm kısım baba bireyden ve geri kalan kısımlar anne bireyden alınarak erkek çocuğun oluşması sağlanır. Kız çocuk elde edilirken aynı prosedür tersinden uygulanmaktadır. Bu prosedürler uygulanırken parti-çalışan ve çalışan-makine kısımlarında (Çizelge 4.3 üç ve dördüncü satır) anne ve baba bireyden alınmaktadır. Literatürde çaprazlama olasılığı genel olarak 0.6 ile 1.0 değerleri arasında seçilmektedir. Büyük olasılık değerleri yerel en iyi değerlere takılma ihtimalini arttırırken küçük olasılık değerleri iyi çözümlere yavaş ulaşılmasına neden olmaktadır (Altay, 2015). Şekil 4.4 uygulanan tek noktalı çaprazlama operatörünü göstermektedir.

Parti listesi	2			4		1		3		
Parti-çalışan atama	1	2	-	1	3	2	-	1	3	-
Çalışan-makine atama	1-2	3-4	-	2-3	1-4	1-2-3-4	-	3-4	1-2	-

Parti listesi	3			1		4		2		
Parti-çalışan atama	1	3	2	1	2	3	-	1	3	2
Çalışan-makine atama	1-2	3	4	1-4	2-3	1-2-3-4	-	3	1-2	4

Pozisyon numarası	1			2		3		4		
Parti listesi	2			3		1		4		
Parti-çalışan atama	1	2	-	1	3	2	1	2	3	-
Çalışan-makine atama	1-2	3-4	-	1-2	3	4	1-4	2-3	1-2-3-4	-

Parti listesi	2			4		1		3		
Parti-çalışan atama	1	2	-	1	3	2	-	1	3	-
Çalışan-makine atama	1-2	3-4	-	2-3	1-4	1-2-3-4	-	3-4	1-2	-

Parti listesi	3			1		4		2		
Parti-çalışan atama	1	3	2	1	2	3	-	1	3	2
Çalışan-makine atama	1-2	3	4	1-4	2-3	1-2-3-4	-	3	1-2	4

Pozisyon numarası	1			2		3		4		
Parti listesi	3			2		4		1		
Parti-çalışan atama	1	3	2	1	2	-	1	3	2	-
Çalışan-makine atama	1-2	3	4	1-2	3-4	-	2-3	1-4	1-2-3-4	-

Şekil 4.4 : Tek noktalı çaprazlama operatörü.

Mutasyon: Çaprazlama operatörünün uygulanması sonucunda elde edilen bireylerin çeşitlendirilmesi amacıyla mutasyon operatörü uygulanmaktadır. Mutasyon operatörü bireyler üzerinde küçük değişimler yaparak daha iyi çözümler elde edilmesi amacıyla kullanılmaktadır. Bu tez kapsamında yer değişim mutasyon operatörü kullanılmaktadır. İlk olarak *pmutation* olasılığı ile rassal olarak iki farklı parti seçilmekte ve daha sonra bu partilerin pozisyonları birbirleri arasında yer değiştirilmektedir. Bu değişim çalışan ve makine atamalarını etkilemektedir ve dolaylı olarak hem parti-çalışan atama hem de çalışan-makine atamaları etkilenmektedir. Bu nedenle her iki kısmın mutasyon sonucunda düzenlenmesi gerekmektedir. Literatürdeki çalışmalara bakıldığında mutasyon olasılığının 0.005 ile 0.02 arasında seçildiği gözlemlenmiştir (Altay, 2015).

5. DENEYSEL HESAPLAMALAR

Genetik algoritma farklı öncelik kuralları ile birlikte uygulandığında performansını değerlendirmek ve aralarında karşılaştırmalar yaparak en uygun öncelik kuralını belirlemek için bu kısımda deneyler gerçekleştirilmiştir. Deneylerde kullanılan veriler gerçek üretim ortamından elde edilen veriler kullanılarak türetilmiştir. Gerçek üretim ortamı verileri Çizelge 5.1 içerisinde gösterilmektedir.

Dört farklı ürün ailesi bulunmaktadır ve her ürün ailesi farklı bir hücreye tahsis edilmiştir. Gruplar oluşturulurken ürünlerin nominal çapları (fiziksel özellikleri) dikkate alınmıştır. Bu ürün aileleri, DN 25-32-40-50, DN 65-80, DN 100-125-150 ve DN 200-250-300 olarak adlandırılmaktadırlar. Her ürün ailesi kendi içerisinde döner flanşlı (DF) ve sabit flanşlı (SF) olmak üzere ikiye ayrılmaktadır.

Toplamda yedi farklı tip makine bulunmaktadır ve bunlar, silindir, TIG kaynak, yatay formlama, dikey formlama, makara, kesme ve kapatma makineleridir. Makinelerde otomatik ve manuel operasyonlar gerçekleştirilebileceği gibi her iki operasyon tipi eş zamanlı olarak gerçekleştirilen makineler de bulunmaktadır.

Çizelge 5.1 : Hücre çevrim süreleri, parti büyüklükleri, operasyon süreleri ve toplam yürüme süreleri.

Çevrim süresi						Operasyon ve yürüme süreleri (sn.) – $v_{z,i,m}$									
Hücreler	Hücreler			(cx_i)	(cn_i)	(q_i)	Silindir	TIG Kaynak	Yatay Form		Dikey Form	Makara	Kesme	Kapatma	(o_i)
Hücre 1	DN 25	1- DF	957	450	5	16	150	KO	KO	450	155	108	62	16	
		2- SF	622	468	4	16	150	110	358	KO	160	108	62	16	
	DN 32	3- DF	915	432	10	15	144	KO	KO	432	148	105	55	16	
		4- SF	590	455	5	15	144	100	355	KO	152	105	58	16	
	DN 40	5- DF	887	420	15	15	140	KO	KO	420	145	96	55	16	
		6- SF	555	438	8	15	140	88	350	KO	145	96	55	16	
	DN 50	7- DF	741	319	15	12	135	KO	KO	319	130	84	45	16	
		8- SF	508	355	5	12	135	75	280	KO	138	84	48	16	
Hücre 2	DN 65	9- DF	626	250	10	9	120	KO	KO	250	117	71	44	15	
		10- SF	436	272	10	9	120	60	212	KO	117	71	44	15	
	DN 80	11- DF	472	155	15	8	108	KO	KO	155	88	66	32	15	
		12- SF	363	186	9	8	108	26	160	KO	102	66	38	15	
Hücre 3	DN 100	13- DF	525	182	12	8	115	KO	KO	182	95	71	40	14	
		14- SF	414	210	8	8	115	50	160	KO	112	71	44	14	
	DN 125	15- DF	617	226	14	11	131	KO	KO	226	102	85	48	14	
		16- SF	469	258	8	11	131	58	200	KO	120	85	50	14	
	DN 150	17- DF	755	318	10	14	145	KO	KO	318	115	94	55	14	
		18- SF	541	360	5	14	145	80	280	KO	134	94	60	14	
Hücre 4	DN 200	19- DF	940	431	5	18	153	KO	KO	431	139	118	65	16	
		20- SF	653	490	15	18	153	120	370	KO	155	118	73	16	
	DN 250	21- DF	1070	485	10	25	168	KO	KO	485	163	135	78	16	
		22- SF	739	545	8	25	168	135	410	KO	175	135	85	16	
	DN 300	23- DF	1202	528	10	34	185	KO	KO	528	188	161	90	16	
		24- SF	857	607	5	34	185	147	460	KO	211	161	103	16	

5.1 Deneysel Veriler

Bu bölümde küçük ve büyük olmak üzere iki farklı problem boyutu ele alınmaktadır. Her iki problem boyutu için gerçek üretim verileri göz önünde bulundurularak deneysel veriler elde edilmiştir.

Küçük boyutlu problemler için 16 tane problem sınıfı üretilmiştir. Her problem sınıfı için çalışan yetenek seviyeleri eş kabul edilerek algoritma performanslarının en iyi sonuçlar ile karşılaştırılması sağlanmıştır. Sistemde bulunan toplam makine başı parti sayısı 40 değerinden daha az olduğunda problem küçük boyutlu daha fazla olduğunda problem büyük boyutlu olarak adlandırılmaktadır.

Büyük boyutlu problemlerin türetilmesi için farklı faktörler göz önünde bulundurularak deney tasarımı oluşturulmuştur. Deney tasarımında toplamda dört faktör değerlendirilmiş ve her faktör için üç seviye belirlenmiştir.

Büyük boyutlu problemler için toplamda 81 ($3 \times 3 \times 3 \times 3$) farklı problem sınıfı üretilmiştir. Çalışan yetenek seviyeleri göz önünde bulundurulduğunda her bir problem tipi 10 farklı örnek elde edilmiştir. Algoritmaların rassallığından kaynaklanan sapmaları azaltmak adına her örnek her algoritma için farklı tohum değerleri ile beş kere çözdürülmüş ve toplamda ($81 \times 10 \times 5 \times 5$) 20250 koşum gerçekleştirilmiştir. Her problem tipi için farklı örnekler elde edilmesinde kullanılan çalışan yetenek seviyeleri Çizelge 5.4 içerisinde gösterilmektedir.

Her örnek için parti büyüklükleri düzgün dağılıma uygun olarak [1, 20] aralığında, manuel operasyon süreleri düzgün dağılımı uygun olarak [1, 50] aralığında, otomatik operasyon süreleri düzgün dağılıma uygun olarak [1,100] aralığında ve hücre içi ortalama yürüme süreleri düzgün dağılıma uygun olarak [1, 4] aralığında rassal olarak belirlenmiştir.

Çizelge 5.2 ve 5.3 sırasıyla küçük ve büyük boyutlu problemlerin elde edilmesinde kullanılan faktörleri ve seviyelerini göstermektedir.

Çizelge 5.2 : Küçük boyutlu problemler için faktörler ve seviyeleri.

Faktörler	Seviyeler	
	1	2
Hücre Sayısı (HS)	2	3
Hücre Parti Sayısı (HPS)	2	4
Hücre Makine Sayısı (HMS)	2	3
Toplam Çalışan Sayısı (TÇS)	3	4

Çizelge 5.3 : Büyük boyutlu problemler için faktörler ve seviyeleri.

Faktörler	Seviyeler		
	1	2	3
Hücre Sayısı (HS)	4	5	6
Hücre Parti Sayısı (HPS)	5	8	10
Hücre Makine Sayısı (HMS)	4	5	6
Toplam Çalışan Sayısı (TÇS)	6	9	12

Çizelge 5.4 : Çalışan yetenek seviyeleri.

	Yüksek	Orta	Düşük
1	33%	33%	33%
2	66%	33%	0
3	66%	0	33%
4	100%	0	0
5	0	66%	33%
6	33%	66%	0
7	0	100%	0
8	33%	0	66%
9	0	33%	66%
10	0	0	100%

Çalışanların yetenek seviyeleri (Yüksek, Orta ve Düşük) manuel operasyon sürelerini etkilemektedir. Örneğin bir çalışanın yetenek seviyesi düşük ise manuel operasyon süreleri 0.63 değerine bölünmektedir. Costa ve diğ. (2014) çalışmasından elde edilen bu değerlere Çizelge 5.5’ de yer verilmektedir.

Çizelge 5.5 : Yetenek seviyelerinin katsayıları.

	Düşük	Orta	Yüksek
Katsayılar	0.63	1	1.29

5.2 Değerlendirme Kriteri

Geliştirilmiş olan alt sınır denkleminde elde edilen yüzdelere büyük olduğu için bu tez kapsamında karşılaştırma analizinde göreceli yüzde sapması (GYS) kullanılmıştır. Büyük boyutlu problemler için GYS, tüm algoritmalarından elde edilen minimum amaç fonksiyonu değeri ($amaç_{min}$) ve her algoritmanın sağlamış olduğu amaç fonksiyonu değeri ($amaç$) kullanılarak denklem 5.1’de yer verildiği şekilde hesaplanmaktadır.

$$GYS (\%) = \left(\frac{amaç - amaç_{min}}{amaç_{min}} \right) \times 100 \quad (5.1)$$

Küçük boyutlu problemler için GYS değeri hesaplanırken $amaç_{min}$ yerine en iyi sonuçlar kullanılmaktadır.

5.3 Parametre Değerlerinin Hesaplanması

Algoritmalarından elde edilen sonuçlar kullanılan parametre değerlerine bağımlı olduklarından dolayı parametre değerlerinin bilinen en iyi kombinasyonunu belirlemek için deneme yanılma yönteminde kullanılmak üzere tekrarlı deneyler gerçekleştirilmiştir (Ramesh ve diğ., 2012).

Deneyler gerçekleştirilirken her seferinde bir parametre değeri değiştirilerek diğer parametreler sabit tutulmuştur. Bilinen en iyi parametre kombinasyonu farklı parametre değerleri üzerinden bir seri deneyin gerçekleştirilmesi sonucu elde edilmiştir. Karşılaştırma analizinde kullanılmak üzere türetilen ve bir önceki kısımda yer verilen deneysel verilerin bir ve üçüncü seviyeleri aynı zamanda parametre değerlerinin belirlenmesinde de kullanılmıştır. Ek olarak, her çalışanın yetenek seviyesi orta kabul edilmiştir.

Her algoritma rassallıktan kaynaklanan varyasyonu göz önünde bulundurmak amacıyla farklı tohum değerleri ile her örnek için beş kez çalıştırılmıştır. Algoritmaların yakınsamalarını sağlamak için her algoritma 30 iterasyon boyunca %1 altında iyileştirme sağlamadığında durdurulmuştur. Bu analiz sonucunda algoritmalar için 10000 çözüm değerlendirmesinin yakınsama sağlamak açısından yeterli olduğu sonucuna varılmış ve adil bir karşılaştırma gerçekleştirmek için her

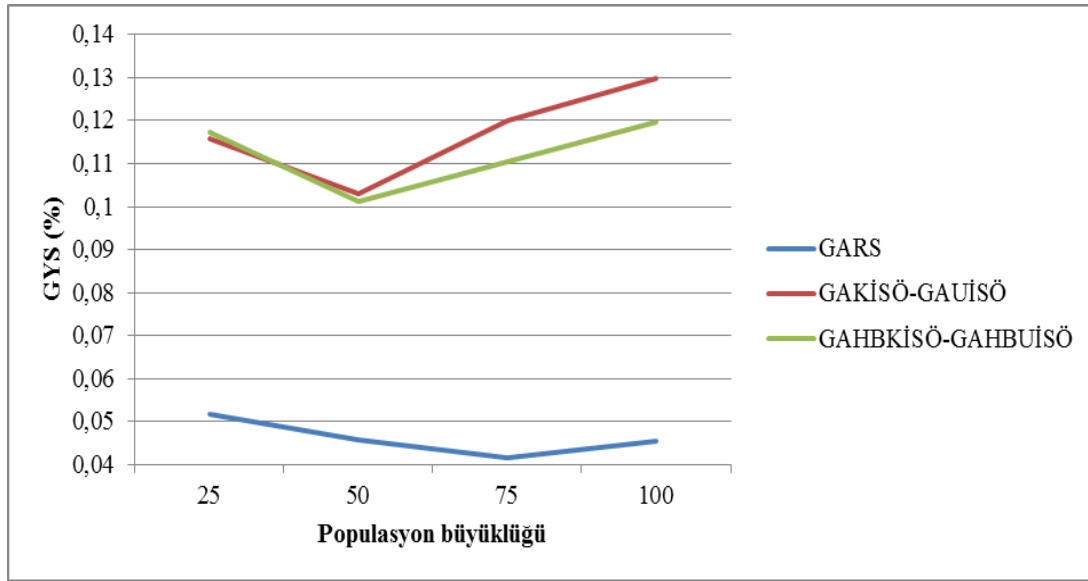
algoritma üzerinden 10000 sonuç deęerlendirilmiřtir. Bu sayede iterasyon sayısı belirlenmiřtir. Toplamda ($16 \times 5 \times 3 \times 48$) 11520 kořum gerekleřtirilmiřtir.

Genetik algorithmada girdi parametresi olarak populasyon byklę, aprazlama olasılıęı ve mutasyon olasılıęı kullanılmaktadır. Bu parametreler iin belirlenen deęerler ve kombinasyonlarından elde edilen ortalama GYS deęerlerine izelge 5.6' da yer verilmektedir. En kk ortalama GYS deęeri kalın olarak gsterilmiř ve algoritmaların kořturulmasında esas olarak alınmiřtır.

Parametre deęerleri belirlenirken GAKİS-GAİS ve GAHBKİS-GAHBİS ikilileri iin ortak deęerler belirlenmiřtir. Bylece toplamda  farklı algoritma iin parametre deęerleri belirlenmiř olmaktadır. GAKİS-GAİS iftinin ve aynı řekilde GAHBKİS-GAHBİS iftinin alıřma prensipleri aynı olduęu iin bu iftlerden bir tanesi iin belirlenen parametre deęerleri dięeri iinde kullanılabilmektedir.

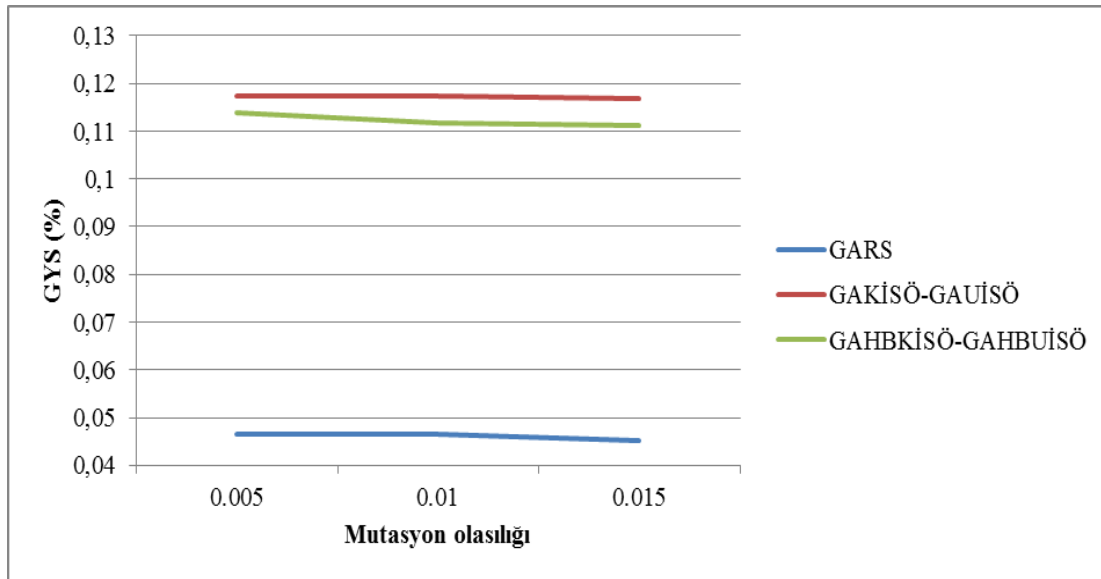
Çizelge 5.6 : Bilinen en iyi parametre değerleri (Genetik Algoritma).

RS	25			50			75			100		
	0.005	0.01	0.015	0.005	0.01	0.015	0.005	0.01	0.015	0.005	0.01	0.015
0.6	0,046	0,05	0,056	0,041	0,046	0,04	0,039	0,042	0,041	0,048	0,044	0,045
0.7	0,051	0,048	0,052	0,047	0,048	0,043	0,045	0,038	0,04	0,046	0,043	0,043
0.8	0,05	0,053	0,049	0,048	0,05	0,044	0,044	0,042	0,042	0,05	0,046	0,044
0.9	0,059	0,055	0,051	0,045	0,051	0,046	0,041	0,04	0,044	0,045	0,047	0,043
KİSÖ- GAÜSÖ												
0.6	0,11	0,106	0,102	0,087	0,093	0,097	0,123	0,106	0,117	0,127	0,132	0,118
0.7	0,108	0,111	0,131	0,102	0,107	0,101	0,11	0,123	0,111	0,125	0,133	0,144
0.8	0,121	0,125	0,113	0,095	0,106	0,114	0,127	0,119	0,132	0,141	0,13	0,129
0.9	0,136	0,112	0,114	0,109	0,119	0,106	0,132	0,12	0,121	0,123	0,136	0,119
HBKİSÖ- HBUİSÖ												
0.6	0,12	0,106	0,102	0,101	0,097	0,104	0,112	0,106	0,128	0,133	0,117	0,114
0.7	0,117	0,111	0,119	0,095	0,091	0,101	0,105	0,11	0,102	0,112	0,126	0,107
0.8	0,134	0,125	0,113	0,103	0,102	0,099	0,113	0,121	0,104	0,121	0,115	0,124
0.9	0,115	0,112	0,133	0,111	0,107	0,104	0,109	0,105	0,112	0,119	0,138	0,111



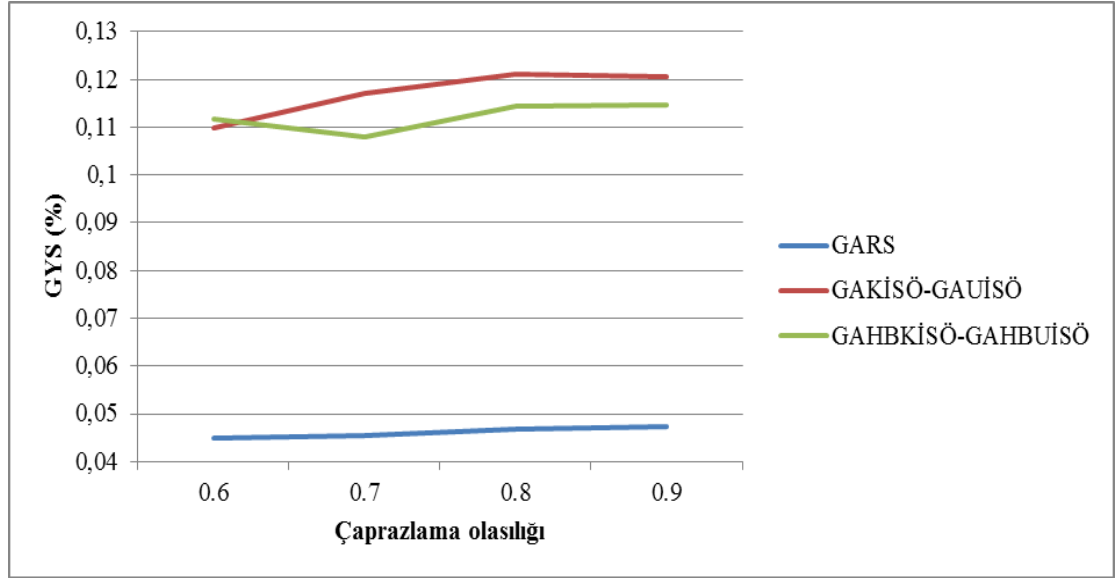
Şekil 5.1 : Populasyon parametresinin GYS üzerine etkisi.

Şekil 5.1 populasyon büyüklüğünün GYS üzerine olan etkisini algoritma performansları üzerinden göstermektedir. Toplamda dört farklı populasyon büyüklüğü algoritmalar için değerlendirilmiştir. GARS algoritması için en iyi sonuç populasyon büyüklüğü 75 değerine eşit olduğunda elde edilirken diğer algoritmalar için en iyi sonuçlar populasyon büyüklüğü 50 değerine eşit olduğunda elde edilmektedir. Bu şekilden GAKİSÖ-GAUİSÖ ve GAHBKİSÖ-GAHBUİSÖ algoritmaları için populasyon büyüklüğü parametresinin etkisinin benzer olduğu sonucuna varılabilmektedir.



Şekil 5.2 : Mutasyon parametresinin GYS üzerine etkisi.

Şekil 5.2 mutasyon olasılığının GYS üzerine olan etkisini algoritma performansları üzerinden göstermektedir. Bu şekilden mutasyon olasılığının GYS değişimi üzerinde etkisi olmadığı anlaşılmaktadır. Ancak bilinen en iyi parametre değerleri belirlenirken genel ortalamalar dikkate alınmıştır. Bu nedenle, GARS ve GAHBKİSÖ-GAHBUIİSÖ algoritmaları için mutasyon olasılığı 0.01 değerine eşit olduğunda en iyi sonuç elde edilirken, GAKİSÖ-GAUİSÖ algoritmaları için mutasyon olasılığı 0.005 değerine eşit olduğunda en iyi sonuçlar elde edilmektedir.



Şekil 5.3 : Çaprazlama parametresinin GYS üzerine etkisi.

Şekil 5.3 çaprazlama olasılığının GYS üzerine olan etkisini algoritma performansları üzerinden göstermektedir. Bu şekilden çaprazlama olasılığının GARS algoritması üzerinde önemli bir etkisi olmadığını ve nispeten GAKİSÖ-GAUİSÖ ve GAHBKİSÖ-GAHBUIİSÖ algoritmaları üzerinde daha önemli bir etkisi olduğu sonucuna varılmaktadır. Bilinen en iyi parametre değerleri belirlenirken genel ortalamalar kullanıldığı için GARS ve GAHBKİSÖ-GAHBUIİSÖ algoritmaları için çaprazlama olasılığı 0.7 değerine eşit olduğunda en iyi sonuçlar elde edilmektedir. GAKİSÖ-GAUİSÖ için çaprazlama olasılığı 0.6 değerine eşit olduğunda en iyi sonuçlar elde edilmektedir.

Şekiller beraber değerlendirildiğinde tüm algoritmaların performansında popülasyon büyüklüğünün etkisinin önemli olduğu sonucuna varılmaktadır. Mutasyon olasılığının algoritma performansları üzerinde ciddi bir etkisi bulunmamaktadır. Çaprazlama olasılığının ise GARS algoritması üzerinde etkisinin bulunmadığı sonucuna varılmaktadır.

5.4 Karşılaştırma Analizi

Algoritmalar arasından farklı senaryolar ışığında en iyi sonucu verenin belirlenmesi için hem küçük hem de büyük boyutlu problemler oluşturulmuş ve koşum sonuçları göz önünde bulundurularak değerlendirilmiştir.

Küçük boyutlu problemler için toplamda 16 farklı sınıf oluşturulmuş ve her sınıf için bir örnek üretilmiştir. Çizelge 5.7 ve 5.8 küçük boyutlu problemler için elde edilen sonuçları göstermektedirler. Problemler belirlenirken hücre sayısı (HS), hücre parti sayısı (HPS), hücre makine sayısı (HMS) ve toplam çalışan sayısı (TÇS) faktörleri ($HS \times HPS \times HMS \times TÇS$) göz önünde bulundurulmuştur.

Çizelge 5.7 elde edilen değerleri ve algoritmaların koşum sürelerini gösterirken Çizelge 5.8 algoritmaların GYS (%) değerlerini göstermektedir. Her problem için en iyi sonuçlar GAMS® 23.5/CPLEX 12.2 paket programı kullanılarak elde edilmiştir.

Çizelge 5.7' den anlaşılabacağı üzere önerilen her algoritma kısa süre içerisinde çözüm sağlamaktadır. Çizelge 5.7 ve 5.8 birlikte değerlendirildiğinde ise algoritmaların kısa süre içerisinde oldukça düşük GYS değerleri elde ettiği görülmektedir. Özellikle GARS algoritmasının bir problem sınıfı hariç tüm sınıflar için en iyi çözüm elde ettiği görülmektedir. Bu sonuçlar üzerinden küçük boyutlu problemler için GARS' nin en iyi sonuçları sağladığı ve diğer algoritmaların birbirlerine benzer olan iyi sonuçlar sağladığı sonucuna varılmaktadır.

Çizelge 5.7 : Küçük boyutlu problemler için elde edilen sonuçlar.

Problemler	Alt sınır	Optimal	Süre	GARS	Süre	GAKİSÖ	Süre	GAUİSÖ	Süre	GAHBKİSÖ	Süre	GAHBUIİSÖ	Süre
1×1×1×1	8132	8220	6.1	8220	3.7	8220	3.7	8220	3.5	8220	3.9	8220	3.5
1×1×1×2	10843	10960	5.9	10960	3.8	10960	3.7	10960	3.5	10960	3.9	10960	3.4
1×1×2×1	9100	9198	23.7	9198	3.7	9198	3.4	9198	3.4	9198	3.5	9198	3.2
1×1×2×2	12135	12264	9.8	12264	3.4	12264	3.7	12264	3.2	12264	3.4	12264	3.4
1×2×1×1	10524	16272	296.6	16272	7.9	16272	7.7	16272	8.4	16272	7.6	16272	8.3
1×2×1×2	14032	21696	495.2	21696	8.3	21696	7.8	21696	7.3	21696	8.0	21696	8.3
1×2×2×1	13315	18207	896.7	18207	8.6	18207	7.9	18207	8.9	18207	8.3	18207	8.5
1×2×2×2	17753	24276	1036.1	24276	8.4	24276	8.0	24276	7.8	24276	8.1	24276	8.4
2×1×1×1	6021	8220	136.4	8220	9.0	8220	9.0	8220	8.8	8220	8.0	8220	8.6
2×1×1×2	8030	10960	14.6	10960	8.2	10960	9.2	10960	8.4	10960	9.3	10960	8.7
2×1×2×1	7428	10736	3229.8	10736	7.5	10736	9.2	10736	8.2	10736	9.4	10736	8.0
2×1×2×2	9904	12265	49.8	12265	8.0	12265	9.0	12265	7.8	12265	9.0	12265	7.8
2×2×1×1	12765	16443	4983.6	16443	25.3	17553	25.0	17316	22.4	16443	21.5	17285	19.7
2×2×1×2	16660	21924	19226.2	21924	23.7	22448	19.6	21924	20.2	21924	19.2	21924	18.9
2×2×2×1	15583	20841	10862.9	21081	24.7	21602	23.4	21418	22.7	22329	19.8	22016	22.0
2×2×2×2	20788	26824	36000	26824	20.6	26824	17.6	28739	19.8	28361	18.2	27862	18.7

Çizelge 5.8 : Küçük boyutlu problemler için algoritma GYS(%) değerleri.

Problemler	GARS	GAKİSÖ	GAUİSÖ	GAHBKİSÖ	GAHBUIİSÖ
1×1×1×1	0,00	0,00	0,00	0,00	0,00
1×1×1×2	0,00	0,00	0,00	0,00	0,00
1×1×2×1	0,00	0,00	0,00	0,00	0,00
1×1×2×2	0,00	0,00	0,00	0,00	0,00
1×2×1×1	0,00	0,00	0,00	0,00	0,00
1×2×1×2	0,00	0,00	0,00	0,00	0,00
1×2×2×1	0,00	0,00	0,00	0,00	0,00
1×2×2×2	0,00	0,00	0,00	0,00	0,00
2×1×1×1	0,00	0,00	0,00	0,00	0,00
2×1×1×2	0,00	0,00	0,00	0,00	0,00
2×1×2×1	0,00	0,00	0,00	0,00	0,00
2×1×2×2	0,00	0,00	0,00	0,00	0,00
2×2×1×1	0,00	0,04	0,05	0,00	0,05
2×2×1×2	0,00	0,02	0,00	0,00	0,00
2×2×2×1	0,02	0,07	0,08	0,07	0,06
2×2×2×2	0,00	0,00	0,03	0,06	0,04
Ortalama	0,00	0,01	0,02	0,01	0,01

Çizelge 5.9 : Büyük boyutlu problemler için algoritma GYS(%) değerleri ve süreler.

Problem	LB	Min	RS	GYS (%)				Süreler				
				GAKİSÖ	GAUİSÖ	GAHBKİSÖ	GAHBUIİSÖ	1	2	3	4	5
1×1×1×1	35083	53025	0,028	0,057	0,059	0,073	0,063	57	53	76	77	79
1×1×1×2	47593	62815	0,034	0,057	0,076	0,076	0,062	58	46	71	68	68
1×1×1×3	66749	84855	0,035	0,059	0,067	0,078	0,067	57	40	66	68	59
1×1×2×1	41987	63920	0,032	0,061	0,081	0,095	0,081	51	56	93	90	83
1×1×2×2	60507	84265	0,029	0,058	0,082	0,081	0,069	49	51	86	80	78
1×1×2×3	69962	90724	0,033	0,056	0,085	0,078	0,069	51	41	74	67	67
1×1×3×1	44242	75198	0,033	0,058	0,093	0,09	0,085	46	57	84	88	81
1×1×3×2	70736	106292	0,035	0,061	0,079	0,094	0,083	47	49	79	76	74
1×1×3×3	79019	114433	0,032	0,057	0,084	0,081	0,079	49	49	73	72	69
1×2×1×1	57476	84089	0,036	0,054	0,065	0,074	0,066	63	105	141	140	135
1×2×1×2	80015	96690	0,031	0,059	0,07	0,078	0,076	62	86	116	117	119
1×2×1×3	110567	134684	0,033	0,057	0,072	0,082	0,069	59	103	114	115	107
1×2×2×1	67320	100170	0,03	0,061	0,078	0,078	0,078	56	104	151	143	134
1×2×2×2	100998	140422	0,033	0,057	0,082	0,085	0,073	55	87	150	132	123
1×2×2×3	123466	168406	0,034	0,055	0,066	0,069	0,071	55	72	144	115	118
1×2×3×1	74441	138330	0,033	0,06	0,07	0,091	0,066	50	107	172	154	149
1×2×3×2	106620	164062	0,034	0,057	0,073	0,081	0,066	50	98	158	137	138
1×2×3×3	149528	213389	0,036	0,063	0,092	0,075	0,065	51	87	146	131	129
1×3×1×1	70271	97833	0,034	0,059	0,078	0,086	0,069	64	125	189	180	164
1×3×1×2	108695	142798	0,034	0,057	0,071	0,08	0,071	62	111	172	151	153
1×3×1×3	159299	193027	0,041	0,063	0,078	0,079	0,073	62	104	155	145	148

Çizelge 5.9 (devam) : Büyük boyutlu problemler için algoritma GYS(%) değerleri ve süreler.

Problem	LB	Min	RS	GYS (%)				Süreler				
				GAKİSÖ	GAUİSÖ	GAHBKİSÖ	GAHBUIÖ	1	2	3	4	5
1×3×2×1	80654	128172	0,029	0,069	0,086	0,091	0,079	56	139	211	180	184
1×3×2×2	129670	195769	0,041	0,066	0,088	0,072	0,077	55	124	220	165	162
1×3×2×3	151762	210763	0,039	0,067	0,084	0,096	0,075	56	110	175	159	160
1×3×3×1	83195	148746	0,035	0,067	0,081	0,087	0,079	52	134	202	194	179
1×3×3×2	139970	216002	0,037	0,069	0,075	0,083	0,072	51	134	187	179	162
1×3×3×3	159596	236445	0,042	0,069	0,075	0,068	0,062	51	115	421	155	159
2×1×1×1	36042	58670	0,045	0,073	0,116	0,107	0,101	107	103	147	156	140
2×1×1×2	50793	71337	0,036	0,065	0,109	0,096	0,106	105	91	136	358	126
2×1×1×3	75758	102727	0,038	0,071	0,084	0,076	0,067	109	74	130	120	349
2×1×2×1	35788	65671	0,031	0,068	0,099	0,118	0,097	92	108	178	165	148
2×1×2×2	61099	87575	0,043	0,058	0,099	0,108	0,088	86	83	141	142	128
2×1×2×3	74951	114127	0,04	0,084	0,096	0,101	0,099	91	80	139	141	125
2×1×3×1	42107	81692	0,036	0,078	0,111	0,091	0,083	88	119	166	163	161
2×1×3×2	64641	104167	0,042	0,075	0,083	0,121	0,11	90	88	157	155	138
2×1×3×3	81833	127517	0,039	0,072	0,106	0,107	0,093	91	97	137	141	117
2×2×1×1	61370	103948	0,033	0,068	0,081	0,086	0,081	113	183	311	264	265
2×2×1×2	82219	122145	0,035	0,069	0,072	0,089	0,081	109	151	272	241	239
2×2×1×3	111059	157764	0,046	0,074	0,061	0,074	0,082	112	148	246	209	206
2×2×2×1	57115	107665	0,043	0,071	0,082	0,118	0,086	108	189	296	296	255
2×2×2×2	83215	128774	0,041	0,076	0,079	0,097	0,083	103	159	280	228	240
2×2×2×3	125678	183519	0,036	0,074	0,087	0,091	0,083	104	154	270	254	231
2×2×3×1	64238	133574	0,039	0,089	0,097	0,093	0,087	94	199	296	298	289
2×2×3×2	102788	169089	0,042	0,081	0,103	0,108	0,107	94	168	262	268	236
2×2×3×3	136688	216776	0,041	0,088	0,094	0,095	0,09	93	160	311	241	243

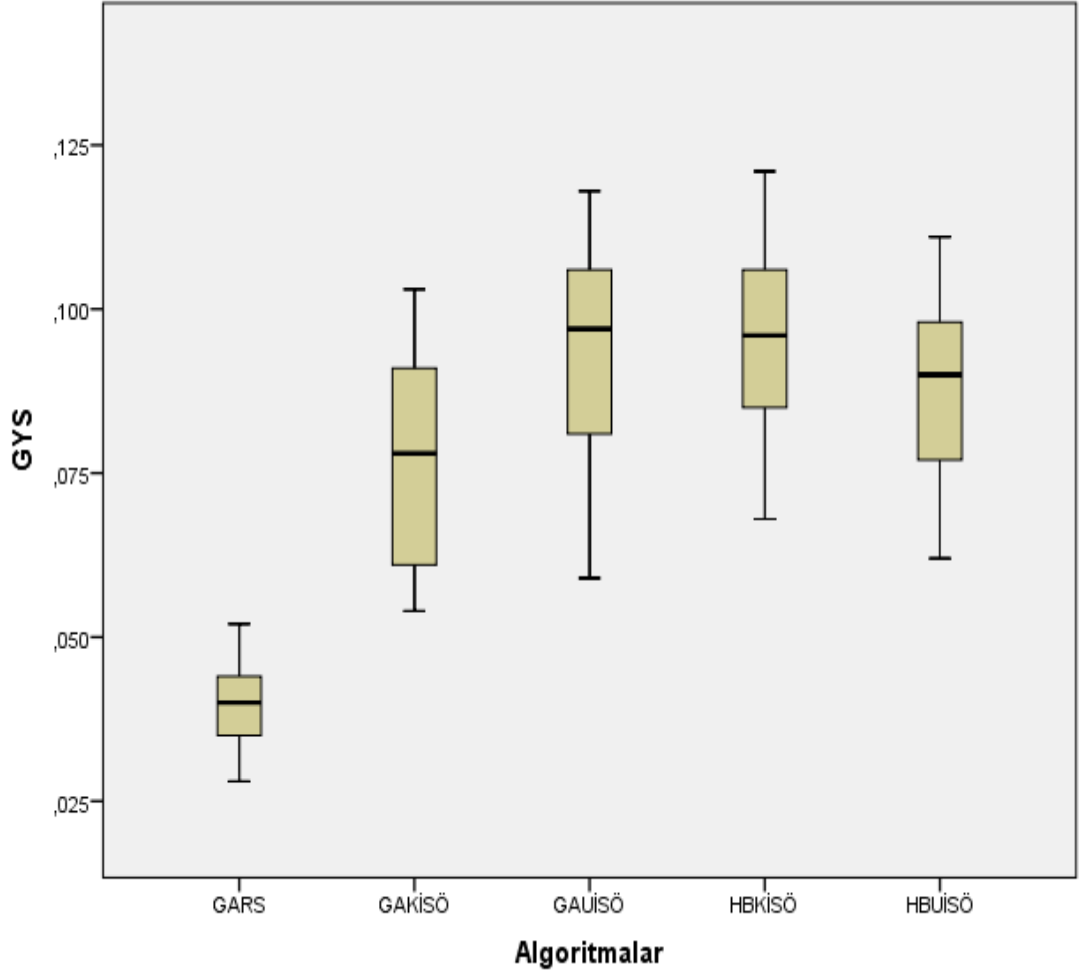
Çizelge 5.9 (devam) : Büyük boyutlu problemler için algoritma GYS(%) değerleri ve süreler.

Problem	LB	Min	RS	GYS (%)				Süreler				
				GAKİSÖ	GAUİSÖ	GAHBKİSÖ	GAHBUIİSÖ	1	2	3	4	5
2×3×1×1	65696	121366	0,045	0,093	0,088	0,088	0,093	117	267	424	353	328
2×3×1×2	102984	152139	0,038	0,089	0,098	0,095	0,092	115	224	338	344	305
2×3×1×3	139265	182917	0,035	0,085	0,093	0,096	0,09	117	175	345	278	270
2×3×2×1	89069	166977	0,044	0,082	0,097	0,098	0,096	108	284	450	372	393
2×3×2×2	115700	184534	0,039	0,089	0,093	0,088	0,086	106	242	351	371	298
2×3×2×3	168187	246523	0,041	0,096	0,097	0,105	0,095	105	226	328	335	328
2×3×3×1	85076	184444	0,048	0,097	0,102	0,103	0,101	96	643	413	363	399
2×3×3×2	108945	187091	0,039	0,088	0,107	0,108	0,092	96	354	389	333	333
2×3×3×3	169377	270900	0,044	0,09	0,103	0,106	0,094	98	231	358	320	290
3×1×1×1	39454	69313	0,043	0,091	0,106	0,117	0,111	148	570	274	256	247
3×1×1×2	56981	87806	0,046	0,09	0,103	0,094	0,102	146	462	231	230	219
3×1×1×3	71221	100463	0,048	0,086	0,109	0,099	0,098	149	439	234	179	187
3×1×2×1	34519	67968	0,047	0,089	0,111	0,111	0,103	120	352	252	270	240
3×1×2×2	60033	103549	0,041	0,098	0,118	0,097	0,103	125	284	253	250	230
3×1×2×3	85890	132456	0,037	0,096	0,113	0,117	0,106	124	501	229	228	213
3×1×3×1	40763	94436	0,045	0,101	0,112	0,104	0,089	114	632	265	274	243
3×1×3×2	54627	106186	0,041	0,1	0,115	0,115	0,106	117	398	2505	250	218
3×1×3×3	85276	134123	0,04	0,089	0,115	0,114	0,105	115	155	221	221	194
3×2×1×1	55354	106463	0,038	0,091	0,098	0,098	0,098	145	328	431	432	403
3×2×1×2	78444	118297	0,044	0,09	0,106	0,098	0,092	147	300	400	389	346
3×2×1×3	120259	175290	0,044	0,092	0,103	0,094	0,089	146	249	381	363	345
3×2×2×1	63402	140487	0,044	0,096	0,102	0,101	0,096	139	351	488	448	430
3×2×2×2	90648	174716	0,043	0,091	0,104	0,107	0,094	145	326	432	463	409
3×2×2×3	122356	202162	0,047	0,099	0,103	0,113	0,095	144	293	437	411	355

Çizelge 5.9 (devam) : Büyük boyutlu problemler için algoritma GYS(%) değerleri ve süreler.

Problem	LB	Min	RS	GYS (%)				Süreler				
				GAKİSÖ	GAUİSÖ	GAHBKİSÖ	GAHBUIİSÖ	1	2	3	4	5
3×2×3×1	68481	166713	0,045	0,091	0,111	0,107	0,096	133	354	546	497	452
3×2×3×2	96436	187910	0,043	0,098	0,115	0,101	0,104	136	357	474	435	409
3×2×3×3	128971	221971	0,044	0,094	0,108	0,105	0,091	129	426	446	407	386
3×3×1×1	68891	139044	0,045	0,099	0,104	0,11	0,099	162	343	644	579	560
3×3×1×2	104731	163674	0,041	0,101	0,109	0,112	0,096	156	265	623	512	448
3×3×1×3	147848	222029	0,051	0,091	0,107	0,096	0,094	156	501	537	475	453
3×3×2×1	68530	144137	0,041	0,1	0,106	0,101	0,107	146	642	629	603	545
3×3×2×2	121365	213379	0,046	0,096	0,106	0,101	0,093	138	417	595	589	529
3×3×2×3	151458	253082	0,052	0,091	0,111	0,098	0,098	142	148	550	522	508
3×3×3×1	90845	224301	0,042	0,103	0,111	0,119	0,106	129	312	763	628	599
3×3×3×2	134995	273682	0,044	0,093	0,105	0,096	0,099	125	261	573	548	549
3×3×3×3	168723	291111	0,046	0,096	0,112	0,112	0,097	130	265	533	489	513
Ortalama	89773	144367	0,039	0,078	0,093	0,095	0,087	98	214	310	262	248

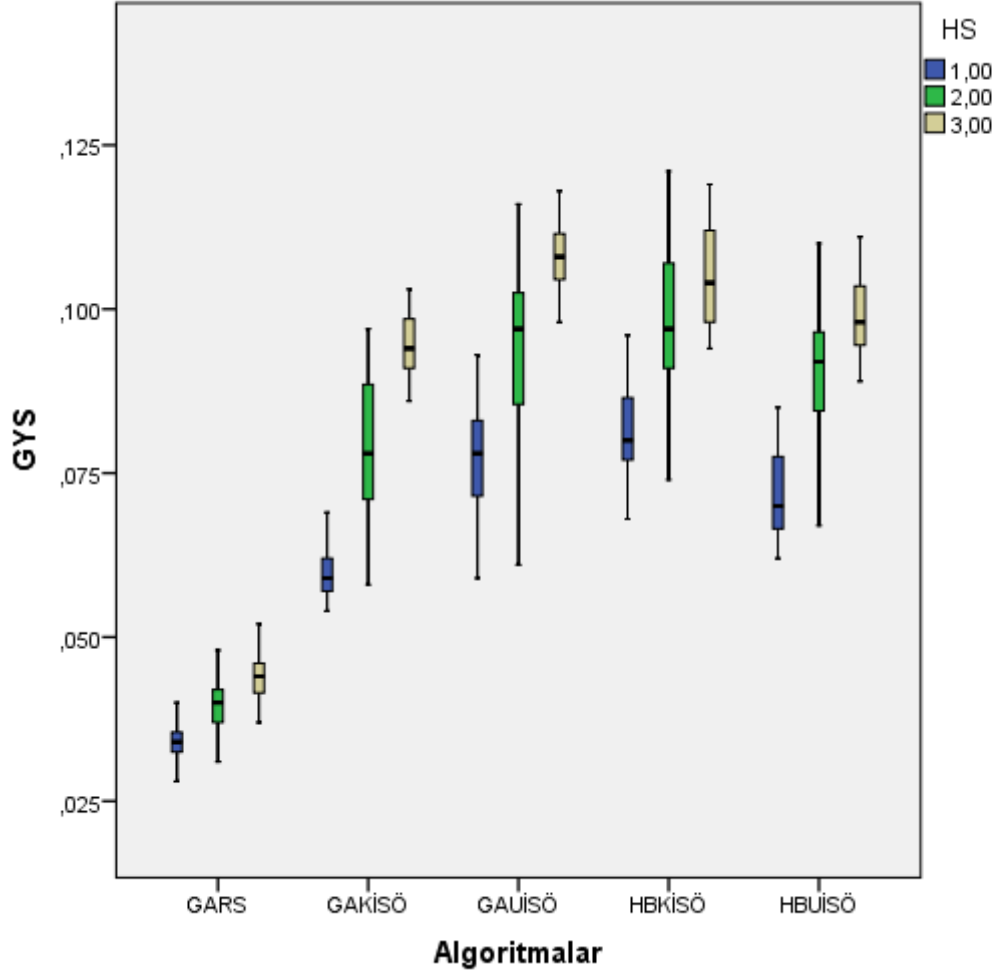
Çizelge 5.9 büyük boyutlu problemler için tüm algoritmalarından sağlanan ortalama GYS (%) değerlerini göstermektedir. Bu değerlere bakıldığında en düşük GYS değerinin GARS algoritması tarafından sağlandığı görülmektedir. Ayrıca GARS algoritması diğer algoritmalara kıyasla daha düşük sürelerde büyük boyutlu problemlere çözüm getirmektedir.



Şekil 5.4 : Algoritmaların GYS değerlerinin 95% güven aralıkları.

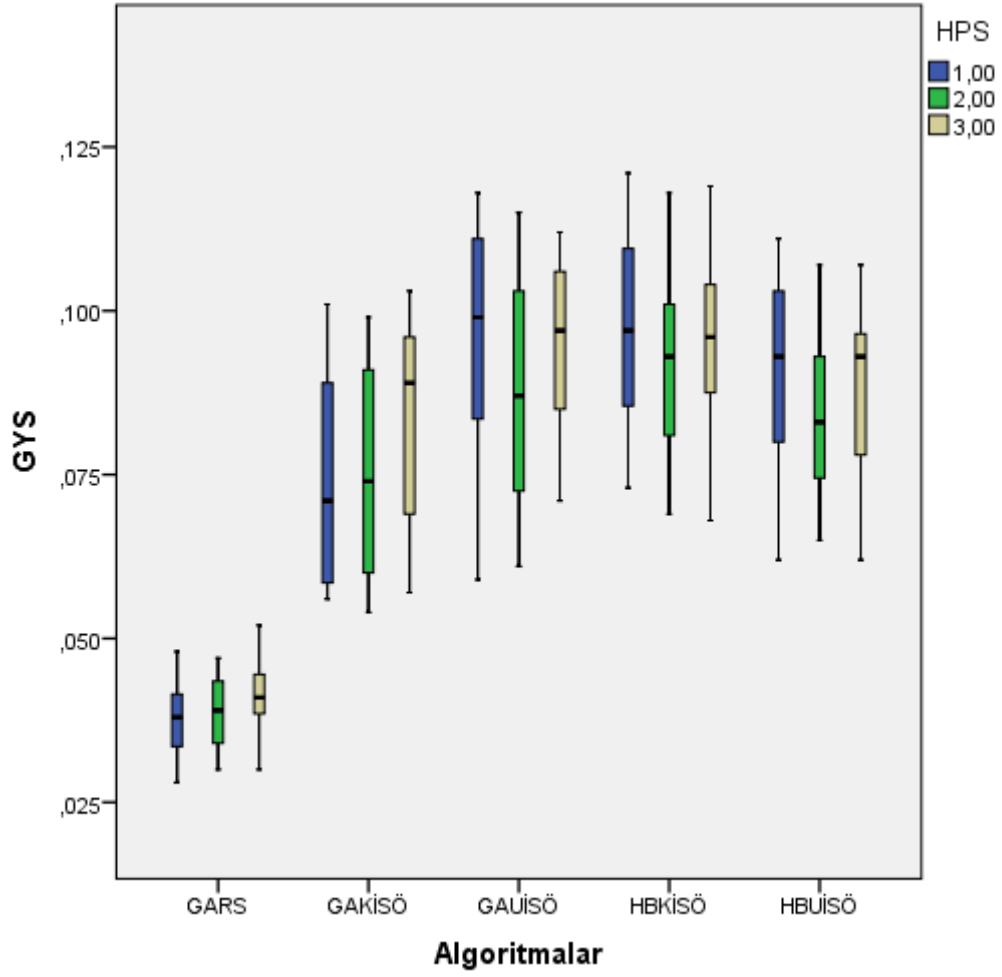
Şekil 5.4 algoritmaların GYS değeri üzerinden güven aralıklarını göstermektedir. Dikkat edildiğinde GARS algoritması diğer algoritmalara nazaran üstün bir performans sergilemektedir. Diğer algoritmalar ise birbirlerine üstünlük sağlayamazken GAKİSÖ ve GAHBUİSÖ algoritmalarının nispeten diğerlerinden daha iyi sonuç verdikleri gözlemlenmiştir. GARS algoritmasının performansının diğerlerine üstünlük göstermesinin nedeni rassallığından kaynaklanmaktadır. Diğer algoritmalar çizelgeleme gerçekleştirirken partilerin belirli kurallar altında çizelgelenmesini sağladıklarından dolayı performansları kısıtlanmaktadır.

Problem sınıfları oluşturulurken kullanılan faktörlerin algoritmaların performansları üzerine olan etkisini incelemek için daha detaylı analizlere ihtiyaç duyulmaktadır. Aşağıda bu analizlere yer verilmiştir.



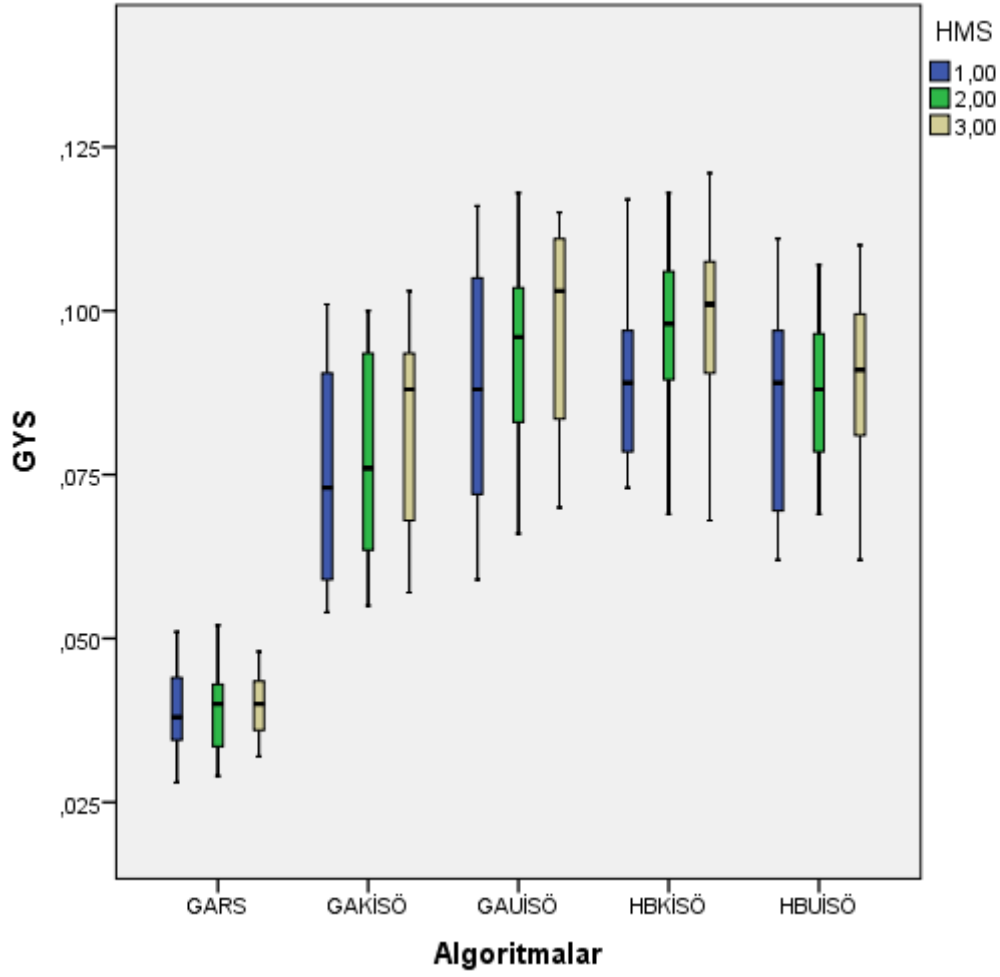
Şekil 5.5 : Algoritmaların hücre sayısı (HS) faktörü üzerinden performansları.

Ortalama GYS değerleri dikkate alındığında GARS algoritmasının diğer algoritmalarından daha iyi sonuçlar verdiği Şekil 5.4’ de görülmektedir. Şekil 5.5 ise GYS değerlerinin HS faktörüne bağımlı olarak değişimini göstermektedir. HS faktörünün seviyesi azaldıkça tüm algoritmaların performanslarında iyileşme gözlemlenmektedir. Bu iyileşme özellikle GAKİSÖ algoritması için daha fazla ön plana çıkmaktadır. Bu grafikten HS faktör seviyesinin değişiminin en fazla GAKİSÖ algoritması üzerinde etkili olduğu sonucu çıkarılmaktadır. En az değişimin ise GARS üzerinde olduğu görülmektedir. Sonuç olarak hücre sayısı arttıkça algoritma performansları üzerinde düşüş gerçekleştiği görülmekte ve HS faktörünün performanslar üzerinde anlamlı bir etkisinin olduğu çıkarılmaktadır.



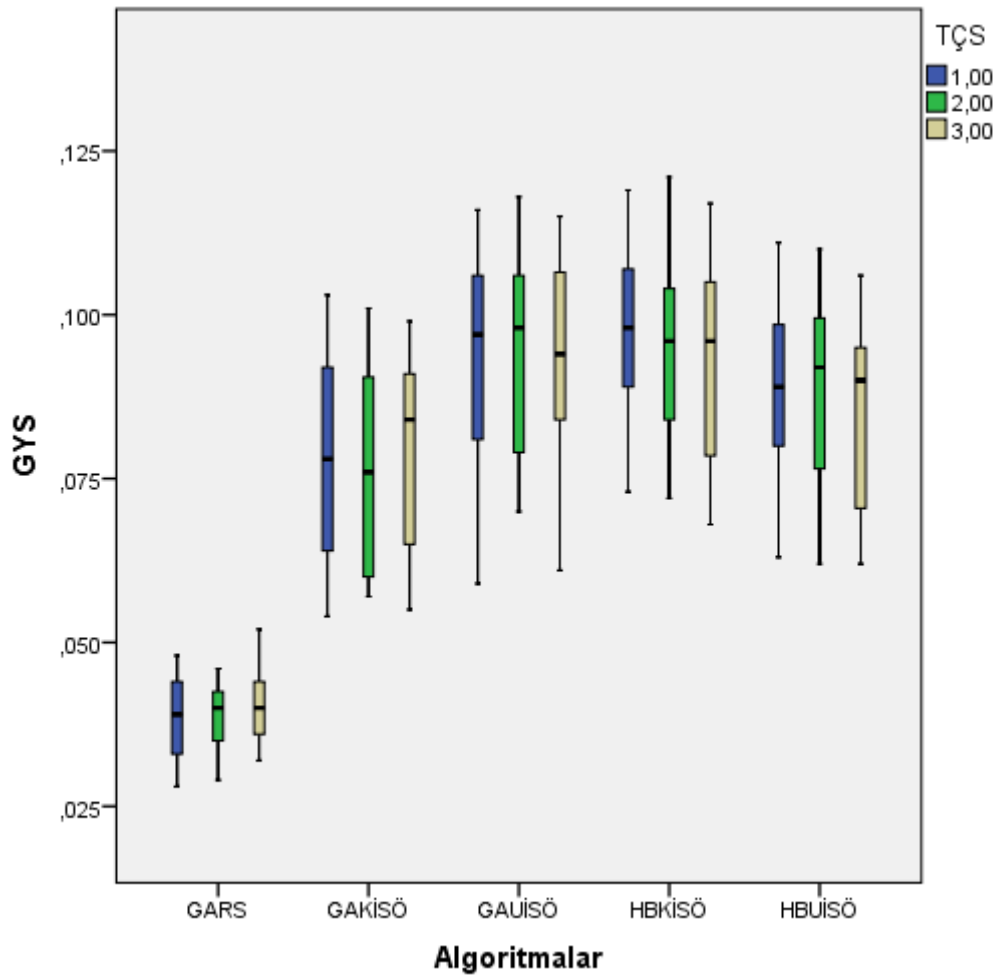
Şekil 5.6 : Algoritmaların hücre parti sayısı (HPS) faktörü üzerinden performansları.

Şekil 5.6 GYS değerlerinin HPS faktörüne bağımlı olarak değişimini göstermektedir. HPS faktörü üzerinden algoritmaların performansları değerlendirildiğinde GARS algoritması hariç tüm algoritmalar için HPS faktör seviyesi ikiye eşit olduğunda (hücre parti sayısı sekize eşit olduğunda) daha iyi sonuçlar verdikleri söylenebilir. Ancak aralıklar birbirleri ile kesişim içerisinde olduğundan faktör seviyeleri arasındaki farkın anlamlı olduğu sonucunu çıkarmak mümkün değildir. Sonuç olarak HPS faktörünün algoritma performansları üzerinde anlamlı bir etkisinin olmadığı sonucuna varılmaktadır.



Şekil 5.7: Algoritmaların hücre makine sayısı (HMS) faktörü üzerinden performansları.

Şekil 5.7' den anlaşılacağı üzere HMS faktörü üzerinden yapılan incelemelerde faktör seviyelerinin algoritma performansları üzerinde önemli bir etkisinin olmadığı görülmektedir. En büyük etki ise HBKİSÖ algoritmasında gerçekleşmiştir. Sonuç olarak, hücrelerdeki makine sayılarının algoritmaların performanslarını etkilemediği ve sonuçlar üzerinde anlamlı bir etkisinin bulunmadığı çıkarılmaktadır.



Şekil 5.8 : Algoritmaların toplam çalışan sayısı (TÇS) faktörü üzerinden performansları.

Şekil 5.8 TÇS faktörünün algoritma performansları üzerine olan etkisini göstermektedir. Performansların değerlendirilmesi açısından yayımlara bakıldığında faktör seviyelerinin anlamlı bir etkisinin olmadığı görülmektedir. Sonuç olarak sistemdeki toplam çalışan sayısının performanslar üzerinde anlamlı bir etkisinin olmadığı sonucuna varılmaktadır.

Yukarıda algoritmaların genel ortalamaları üzerinden yapılan değerlendirmelere yer verilmektedir. Bu değerlendirmeler sonucunda problem sınıflarının oluşturulmasında kullanılan faktörlerden yalnızca bir tanesinin seviye değişiminin performanslar üzerinde etkili olduğu görülmektedir. Sistemdeki toplam hücre sayısı değiştiğinde algoritma performansları anlamlı olarak değişim göstermektedir. Diğer faktörlere bakıldığında algoritmalarla göre kısmi değişimler olduğu gözlemlensede anlamlı bir değişim olduğu sonucu çıkarılamamaktadır.

Genel olarak aşağıdaki sonuçlar çıkarılmıştır.

- Genel ortalamalar değerlendirildiğinde en iyi performansı GARS algoritması göstermektedir ve diğer algoritmalara üstünlük sağlamaktadır.
- GAKİSÖ ve GAHBÜİSÖ algoritmaları performans açısından benzerdir ve diğer iki algortmadan nispeten daha iyi performans göstermektedirler.
- HS faktörünün seviyelerinin algoritmaların performansları üzerinde anlamlı etkisi bulunmaktadır. Hücre sayısı azaldıkça algoritma performansları artmaktadır.
- HPS, HMS ve TÇS faktörlerinin algoritma performansları üzerinde anlamlı etkisi bulunmamaktadır.

Algoritma performansları değerlendirilirken genel ortalamalar dikkate alınmıştır. Tüm çalışan havuzlarının ortalaması üzerinden yapılan değerlendirmelerde GARS algoritması en iyi sonuçları vermektedir. Ancak hangi çalışan havuzunda daha iyi sonuç verdiğini değerlendirebilmek için ayrıca analizlere ihtiyaç duyulmaktadır.

Çizelge 5.10 çalışan havuzlarının maliyet açısından karşılaştırmasını göstermektedir. Maliyet açısından ilk sırayı dokuzuncu havuz almaktadır. Bu havuz 66% yetenek seviyesi düşük ve 33% yetenek seviyesi orta çalışanlardan oluşmaktadır. Maliyet olarak ikinci sırada 10. havuz bulunmaktadır. Bu havuz tamamen yetenek seviyesi düşük çalışandan oluşmaktadır. Maliyet olarak üçüncü sırada beşinci havuz bulunmaktadır. Bu havuz 33% yetenek seviyesi düşük ve 66% yetenek seviyesi orta çalışandan oluşmaktadır. En yüksek maliyetli havuzlara baktığımızda ise dördüncü havuzun ilk sırada yer aldığı görülmektedir. Bu havuz 100% yetenek seviyesi yüksek çalışanlardan oluşmaktadır. Dokuzuncu sırada iki numaralı havuz bulunmaktadır. Bu havuz 33% yetenek seviyesi orta 66% yetenek seviyesi yüksek çalışanlardan oluşmaktadır. Sekizinci sırada üç numaralı havuz bulunmaktadır. Bu havuz ise 33% yetenek seviyesi düşük 66% yetenek seviyesi yüksek çalışanlardan oluşmaktadır.

Çizelge 5.10 : Çalışan havuzlarının maliyetleri

Havuzlar	LB	Min	GARS	Sıralama
1	101947	161591	168121	6
2	124990	178364	185520	9
3	112072	172901	179653	8
4	135115	187862	195242	10
5	91822	150518	156672	3
6	114865	168628	175515	7
7	104740	158529	164954	5
8	89029	155564	161882	4
9	72375	130761	135862	1
10	65986	132170	137642	2

Bu sonuçlar değerlendirildiğinde yetenek seviyesi düşük çalışanların olduğu havuzların daha düşük maliyete ve yetenek seviyesi yüksek çalışanların olduğu havuzların daha yüksek maliyetlere neden oldukları anlaşılmaktadır. Bunun nedeni, yetenek seviyesi yüksek çalışanların birim zaman maliyetlerinin yüksek olmasıdır. Yetenek seviyesi yüksek çalışanlar için belirlenen birim zaman maliyeti 1.29 iken yetenek seviyesi düşük çalışanlar için 0.63 olarak belirlenmiştir. Ancak yetenek seviyesi yüksek çalışanlar maliyetteki artış oranından daha az oranda işlem sürelerinde azalma sağladıkları için toplam maliyet artış göstermektedir. Bunu önlemenin yolu ise yetenek seviyesi yüksek çalışanların birim zaman maliyetlerinin dengelenmesidir. Elde edilen sonuçlar farklı yetenek seviyelerindeki çalışanların toplam maliyet üzerinde oldukça etkili olduğunu göstermektedir.

Sonuç olarak mevcut çalışan havuzları dikkate alındığında GARS algoritmasının dördüncü havuz ile birlikte en iyi sonuçlara ulaştığı görülmektedir.

6. ÖNERİLEN YÖNTEMBİLİM İLE GERÇEK BİR UYGULAMA

Önerilen yöntembilimin uygulanması İstanbul Hadımköy’de bulunan Ayvaz A.Ş. fabrikasında gerçekleştirilmiştir. Ayvaz A.Ş. sekiz farklı ürün grubu üretimi gerçekleştiren bir işletmedir. Bu ürün grupları şunlardır: Körük grubu, Esnek metal hortum grubu, Seviye kontrol grubu, Kondenstop grubu, Vana grubu, Yalıtım grubu, Yangın grubu ve ARI-Armaturen grubu. Bu tez kapsamında başlıca üretimlerden birisi olan körük grubu ele alınmaktadır. Körük üretim tesisi HÜS yapısına sahiptir ve dört bağımsız melez üretim hücresinden oluşmaktadır.

Döner ve sabit flanşlı olmak üzere iki farklı körük çeşidi üretilmektedir. Her iki grup için ürün aileleri ürünlerin çaplarına göre belirlenmiştir. Dört farklı ürün ailesi bulunmaktadır ve her ürün ailesi farklı bir hücreye tahsis edilmiştir. Gruplar oluşturulurken ürünlerin nominal çapları (fiziksel özellikleri) dikkate alınmıştır. Bu ürün aileleri, DN 25-32-40-50, DN 65-80, DN 100-125-150 ve DN 200-250-300 olarak adlandırılmaktadırlar. Ürünlerin işlem gördükleri makineler (operasyonlar) için manuel ve otomatik işlem sürelerine kısım 5 çizelge 5.1’ de yer verilmektedir. Şekil 6.1 körük üretim hücrelerini göstermektedir.



Şekil 6.1 : Körük üretim hücreleri.

Çizelge 5.1 içerisinde 24 farklı ürün parti büyüklükleri ile birlikte gösterilmektedir. Parti büyüklüğü taleple eş değer olarak kabul edilmiştir.

Çizelge 6.1 üç farklı çalışan havuzu için önerilen yöntembilim uygulandığında elde edilen sonuçları göstermektedir. Ele alınan havuzlar toplam çalışan maliyeti açısından değerlendirildiğinde en düşük olan dokuzuncu havuz, orta sırada olan yedinci havuz ve en yüksek olan dördüncü havuzdur. Önerilen yöntembilim üç farklı çalışan havuzu ile uygulanarak elde edilen sonuçlara çizelge 6.2’ de yer verilmiştir. Çizelgeleme periyodu için dokuz saatlik bir çalışma vardiyası ele alınmıştır. Tüm süreler saniye cinsinden olduğu için dokuz saatlik vardiyanın karşılığı olan 32400 saniye çizelgeleme zaman limiti olarak kullanılmıştır. Bu limit aynı zamanda kapasite limiti olarak değerlendirilmekte ve önerilen yöntembilimin birinci aşamasında PS problemi çözülürken sisteme salınan partiler bu limit üzerinden belirlenmektedir.

Çizelge 15 içerisindeki veriler dikkate alındığında dördüncü havuzun α oranı 0.3 değerine eşit olduğunda çizelgeleme periyodunu sağladığı görülmektedir. Toplamda üç farklı parti (2-8-24) parti salınım matematiksel modeli sonuçları dikkate alınarak çizelgelemeye katılmamıştır. Toplam 207 adet parça 365750 maliyetle üretilebilmektedir. Yedinci havuzun α oranı 0.7 değerine eşit olduğunda çizelgeleme periyodunu sağladığı görülmektedir. Toplamda dört farklı parti (2-5-23-24) parti salınım matematiksel modeli sonuçları dikkate alınarak çizelgelemeye katılmamıştır. Toplam 187 adet parça 288648 maliyetle üretilebilmektedir. Dokuzuncu havuzun α oranı 2 değerine eşit olduğunda çizelgeleme periyodunu sağladığı görülmektedir. Toplamda sekiz farklı parti (1-5-7-16-17-19-21-23) parti salınım matematiksel modeli sonuçları dikkate alınarak çizelgelemeye katılmamıştır. Toplamda 143 adet parça 209875 maliyetle üretilebilmektedir. Bu sonuçlardan uygulanan çalışan havuzu politikalarının maliyetler üzerinde önemli etkileri olduğu anlaşılmaktadır.

Çizelge 6.1’ de yer verilen sonuçların özet değerlendirmesine çizelge 6.2’ de yer verilmektedir. Çizelge 6.2 uygulanan çalışan havuzu politikasına göre birim parça maliyetlerini ve sürelerini göstermektedir. En düşük parça başı maliyet değerine dokuzuncu havuz uygulandığında ulaşılmaktadır. En düşük parça başı işlem süresi değerine dördüncü havuz uygulandığında ulaşılmaktadır. Dokuzuncu havuz incelendiğinde bu havuzdaki çalışanların yoğunluğunu yetenek seviyesi düşük çalışanların oluşturduğu görülmektedir. Orta maliyet değerlerine ise yedinci havuzla ulaşılmaktadır. Bu havuzda çalışanların tümü orta yetenek seviyesinde olan çalışanlardır. En yüksek parça başı maliyet değerine ise dördüncü havuz kullanıldığında ulaşılmaktadır. Bu havuzdaki çalışanların tümü yetenek seviyesi yüksek çalışanlardır. Bu sonuçlar değerlendirildiğinde en iyi sonuçların yetenek seviyesi düşük çalışanların yoğunlukta olduğu ancak yetenek seviyesi orta çalışanlara da yer verilen dokuzuncu havuzdan elde edildiği anlaşılmakta ve bu havuz önerilmektedir. Ancak unutulmamalıdır ki bu sonuçlar belirlenen girdi parametrelerine bağlıdır. Çalışanların maliyetleri katsayılarına çizelge 5.5 içerisinde yer verilmiştir. Yetenek seviyesi yüksek ve düşük çalışanların maliyetleri birbirleri ile ters orantılı olacak şekilde belirlenmiştir. Bu sonuçlar bize yetenek seviyesi yüksek çalışanların hücre çevrim süreleri üzerindeki etkisinin maliyet üzerine olan etkisinden daha az olduğunu göstermektedir. Bu nedenle eğer yetenek seviyesi yüksek çalışanlar kullanılacaksa birim zaman maliyetlerinin azaltılması

gerekmektedir. Bu sayede yetenek seviyesi yüksek ve düşük çalışanlar arasındaki fark dengelenebilecektir. Yetenek seviyesi yüksek çalışanların maliyet artışı ancak çevrim sürelerinde ciddi anlamda azalış meydana getirebilirlerse sönmölenebilecektir.

Çizelge 6.1 : Üç farklı havuz ve önerilen yöntem bilim uygulama sonuçları.

Yöntembilim ilk aşama			GARS			
	Salınanlar	Amaç	Maliyet	Yayılma	Süre	α
Havuz 9	Tüm işler	221	374046	55169	68.03	0
Havuz 7	Tüm işler	221	405414	45046	76,2	0
Havuz 4	Tüm işler	221	408916	35221	64,7	0
Havuz 9	2 hariç	217	369076	54436	63	0.1
Havuz 7	2 hariç	217	391293	43477	74.97	0.1
Havuz 4	2 hariç	217	400510	34497	77.29	0.1
Havuz 9	4-24 hariç	211	352696	52020	72.08	0.2
Havuz 7	4-24 hariç	211	374139	41571	74.27	0.2
Havuz 4	4-24 hariç	211	378173	32573	76.54	0.2
Havuz 9	2-8-24 hariç	207	345922	51021	64.46	0.3
Havuz 7	2-8-24 hariç	207	355437	39493	61.40	0.3
Havuz 4	2-8-24 hariç	207	365750	31503	74.52	0.3
Havuz 9	1-2-4-22 hariç	199	327162	48254	71.82	0.4
Havuz 7	1-2-4-22 hariç	199	330030	36670	63.46	0.4
Havuz 9	2-3-23 hariç	197	318158	46926	77.71	0.5
Havuz 7	2-3-23 hariç	197	316197	35133	68.20	0.5
Havuz 9	1-3-23 hariç	196	312612	46108	66.08	0.6
Havuz 7	1-3-23 hariç	196	311148	34572	73.19	0.6
Havuz 9	2-5-23-24 hariç	187	275458	40628	75.80	0.7
Havuz 7	2-5-23-24 hariç	187	288648	32272	65.88	0.7
Havuz 4	2-5-23-24 hariç	187	349798	30129	64.74	0.7
Havuz 9	3-5-18-19-23 hariç	179	271546	40051	57.59	1.0
Havuz 9	2-3-5-17-21-23 hariç	162	240798	35516	62.05	1.5
Havuz 9	1-5-7-16-17-19-21-23 hariç	143	209875	30955	40.08	2.0

Çizelge 6.2 : Çalışan havuzları özet değerlendirme.

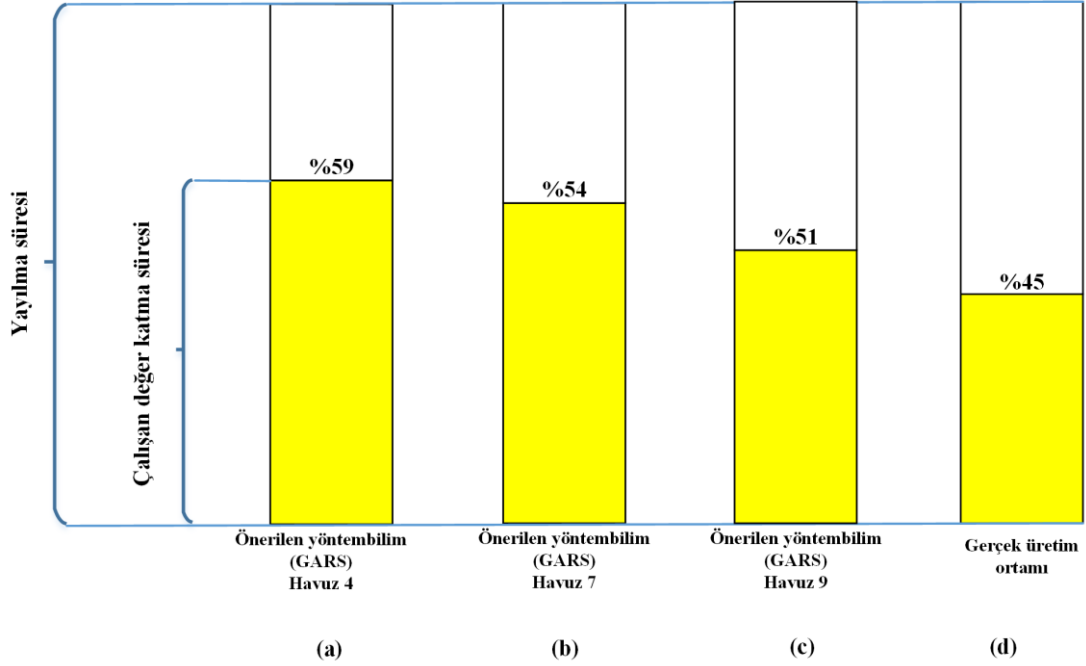
	Maliyet	Parça Sayısı	Parça başı maliyet	Parça başı süre
Havuz 4	365750	207	1767	152
Havuz 7	288648	187	1544	173
Havuz 9	209875	143	1468	216

Farklı havuz stratejileri için çalışanların verimlilik oranları ölçebilmek amacıyla denklem 6.1 geliştirilmiştir. Bu denklem çalışanların ortalama çalışma sürelerinin

yayılma süresi içerisindeki oranını göstermektedir ve çalışan değer katma oranı (ÇDKO) olarak adlandırılmaktadır.

$$\text{ÇDKO} = \left(\left(\sum_{z=1}^Z p_i \times w_{1,z,i} / Z \right) / C \right) \times 100 \quad (6.1)$$

ÇDKO oranı farklı çalışan havuzlarına karakteristiklerine göre değişim göstermektedir. Şekil 6.2 önerilen yöntem bilimin uygulanması sonucunda elde edilen sonuçları göstermektedir.

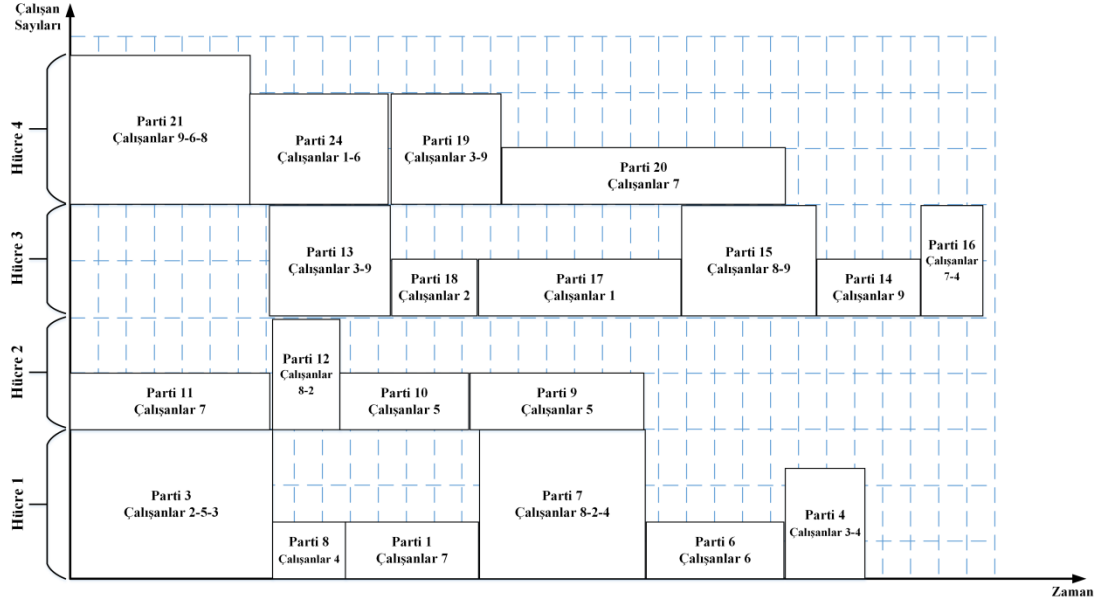


Şekil 6.2 : Çalışan değer katma oranlarının (ÇDKO) havuzlara göre değişimi.

ÇDKO oranları göz önünde bulundurulursa çizelge 6.2’ de verilen değerlerle ters orantılı olduğu gözlemlenmektedir. En yüksek ÇDKO oranına dördüncü havuz kullanılarak ulaşılırken en düşük ÇDKO oranına dokuzuncu havuz kullanıldığında ulaşılmaktadır. Bu durumun nedeni çizelgelenen parti sayısı azaldıkça çizelgeleme sürecinde partilere atanan çalışan sayısının azalmasıdır. Ancak önerilen yöntem bilim hangi çalışan havuzu ile uygulanırsa uygulansın gerçek üretim ortamından elde edilen ÇDKO oranından daha iyi sonuçlar vermektedir.

Ayvaz A.Ş nin körük üretim tesisinden elde edilen gerçek veriler ile uygulanan bir vardiyalık çizelgeleme sonuçlarına şekil 6.3’ de yer verilmektedir. Sistemde toplamda dokuz çalışan bulunmaktadır ve tüm çalışanların orta yetenek seviyelerine sahip oldukları (Havuz 7) dikkate alınmıştır. Bir vardiyanın dokuz saat olduğu göz

önünde bulundurulduğunda yer verilen çizelge 33 eşit 1000 saniyeden oluşan parçalara ayrılarak oluşturulmuştur.



Şekil 6.3 :Önerilen yöntem bilim uygulaması ile elde edilen çizelge (Havuz 7).

Önerilen yöntem bilim uygulanırken ilk olarak α oranı 0 olarak kabul edilmiştir. Bu durumda tüm partilerin (24 adet) sisteme salınımı gerçekleşmiştir. Ancak sisteme salınan partilerin ele alınan vardiya süresi (32400) içerisinde tamamlanmaları mevcut imkanlarla mümkün olmadığı için α oranı 0.1 arttırılmıştır. Bu yöntemin altı kere uygulanması sonucunda α oranı 0.7 değerine eşit olmuş ve olurlu bir çizelge elde edilmiştir. Bu çizelgeden elde edilen ÇDKO oranı yaklaşık olarak 54%' e eşittir. Sistemin gerçek ÇDKO oranının 45%' e eşit olduğu göz önünde bulundurulduğunda uygulanan yöntem bilimin oldukça iyi sonuçlar verdiği ve yaklaşık olarak 20% iyileştirme sağladığı görülmektedir.

Sonuçlar üretim sistemindeki çalışanların hepsinin orta yetenek seviyesinde olduğu varsamıyla elde edilmiştir. Eğer yetenek seviyeleri yüksek olarak kabul edilirse bu durumda çalışan verimlilikleriyle birlikte toplam çalışan maliyetleri de artmaktadır. Benzer şekilde yetenek seviyeleri düşük olarak kabul edilirse bu durumda çalışan verimlilikleriyle birlikte toplam çalışan maliyetleri de azalmaktadır. Bu durum bize maliyetler ve verimlilik arasında bir ikilem olduğunu ve karar mekanizmalarında üzerinde önemle durulması gerektiğini göstermektedir.

Sonu olarak hem alıřan verimlilik oranlarının hem de toplam alıřan maliyetlerinin mevcut alıřan havuzlarından olduka etkilendikleri gzlemlenmiřtir. Bu durum uygulanan alıřan havuz politikalarının nemini ortaya koymaktadır.

7. SONUÇLAR VE DEĞERLENDİRMELER

Bu tez kapsamında ÇKK HÜS' de PS ve PÇÇA problemleri ele alınmıştır. PS probleminin çözülmesiyle işyükü kontrolü sağlanırken PÇÇA problemi ile birlikte işyükünün çizelgelenmesi sağlanmaktadır. İşyükü çizelgelenmesinden kasıt çalışanların ve partilerin eş zamanlı olarak çizelgelenmesidir. Ele alınan sistemde periyodik salınım gerçekleştirilmekte ve belirli zaman kısıtı altında partilerin sisteme salınımı sağlanmaktadır. Aşırı miktarda partinin sisteme salınması olurlu çizelgelerin elde edilmesini önlerken düşük miktarda salınım ise verimsizliklere neden olmaktadır. Bu nedenle uygun salınım miktarlarının belirlenmesi gerekmektedir. Sisteme salınan partilerin belirli bir zaman aralığında çalışan atamalarının da gerçekleştirilerek çizelgelenmesi için PÇÇA problemi tanıtılmıştır. Bu problemin çözülmesi ile birlikte partilerin hücrelerdeki operasyonlarına hangi zaman aralıklarında hangi çalışanların atandıkları belirlenebilmektedir.

Yukarıdaki bilgiler değerlendirildiğinde hem sisteme salınan partilerin belirlenmesi ve işyükü kontrolü için ortaya koyulan PS probleminin hem de salınan partilerin çalışanlarla birlikte çizelgelenmesi için ortaya koyulan PÇÇA problemlerinin birbirleriyle ilişkili oldukları görülmektedir. Bu nedenle bu tez kapsamında her iki problem geliştirilmiş olan bir yöntembilim altında aşamalı olarak birleştirilmiştir. Yöntembilim kullanılarak ilk olarak PS problemine çözüm aranmakta ardından PS problemi için bulunan çözüm PÇÇA problemi için girdi olarak kullanılarak probleme çözüm aranmaktadır. Her iki problemin net olarak ifade edilmesi ve çözüm aranması amacıyla matematiksel modeller geliştirilmiştir. PS problemi için geliştirilen matematiksel modelde amaç sisteme salınan parça sayısının en büyüklenmesi olarak belirlenirken PÇÇA problemi için geliştirilen matematiksel modelde amaç toplam çalışan maliyetinin en küçüklenmesidir. Bu sayede hem yayılma süresi hem de çalışan maliyetleri eş zamanlı olarak düşürülebilmektedir.

Önerilen yöntembilimin uygulanması için Ayvaz A.Ş. körük üretim tesisi seçilmiştir. Bu üretim tesisinin birbirinden bağımsız birden çok melez üretim hücresinden oluştuğu ve çift kaynak kısıtlı (ÇKK) bir sistem olduğu gözlemlenmiştir. Sistemin ÇKK olmasının nedeni çıktı sayılarının hem hücrelere atanan çalışanların

kapasitelerine hem de mevcut hücre kapasitelerine bağımlı olmasıdır. Bu nedenle sistem çift kaynak kısıtlı hücresel üretim sistemi (ÇKK HÜS) olarak adlandırılmıştır. Melez üretim hücrelerinde hem manuel hem de otomatik operasyonlar gerçekleştirilebilmektedir. Hücrelerdeki operasyonlara atanan çalışanların yetenek seviyeleri ve sayıları hücre çevrim süresini direkt olarak etkilemektedir. Böylece parti işlem süreleri ve hücre çıktı miktarları dolaylı olarak etkilenmektedir. Bu etkiler dikkate alınarak tek parça akışa, çalışan yetenek seviyelerine ve çalışan uygunluklarına PÇÇA problemi için geliştirilen matematiksel modelde yer verilmiştir.

Önerilen yöntem bilimin ilk aşamasında yer verilen PS problemi için geliştirilen matematiksel modelin en iyi çözümlerine polinomsal süreler içerisinde ulaşılabildiğinden çözüm için GAMS® 23.5/CPLEX 12.2 paket programı kullanılmıştır. Ancak ikinci aşamada yer verilen PÇÇA problem için geliştirilen matematiksel modelin çözümüne polinomsal süre içerisinde ulaşamadığından çözüm için meta-sezgisel yöntemler kullanılmıştır. Kullanılan meta-sezgisel yöntemler için bilinen en iyi parametre değerlerini bulabilmek için deneme yanılma yönteminde kullanılmak üzere bir seri deney gerçekleştirilmiş ve elde edilen sonuçlardan bilinen en iyi parametre değerleri elde edilmiştir.

PÇÇA problemi için küçük ve büyük boyutlu iki farklı problem boyutu belirlenmiştir. PÇÇA problemi için deney tasarımı kullanılarak 16 farklı küçük boyutlu problem tipi ve 81 farklı büyük boyutlu problem tipi belirlenmiştir. Ayrıca çalışan yetenek seviyelerinin sistem performansı üzerine olan etkisini ölçebilmek amacıyla 10 farklı çalışan havuzu belirlenmiştir. Bu havuzlar belirlenirken üç farklı yetenek seviyesi göz önünde bulundurularak çalışanların farklı oranlarda bu yetenek seviyelerine atanması gerçekleştirilmiştir.

PÇÇA problemine çözüm bulabilmek için GA ile birlikte beş farklı öncelik kuralı kullanılmış ve her öncelik kuralı farklı bir algoritma olarak değerlendirilerek küçük ve büyük boyutlu problemler üzerinden karşılaştırılmaları sağlanmıştır. Ayrıca büyük boyutlu problemlerin üretilmesi için kullanılan deney tasarımında yer verilen faktörlerin etkileri kısım 5 içerisinde detaylı olarak analiz edilmiştir.

Aşağıda öncelik kuralları ve faktörler için elde edilen sonuçların özetine yer verilmektedir.

- Algoritma performansları üzerinde etkisi en yüksek olan parametre popülasyon büyüklüğüdür.
- Tüm meta-sezgisel yöntemler en iyi çözüm süreleri ile kıyaslandığında kısa sürelerde olurlu çözümlere ulaşmaktadırlar.
- Geliştirilen öncelik kuralları kullanıldığında en iyi sonuçların RS kuralı ile birlikte elde edildiği gözlemlenmiştir. Bu nedenle en iyi algoritma GARS olarak belirlenmiştir.
- GAKİSÖ algoritmasının GARS algoritmasından kötü ancak diğer algoritmalarından daha iyi sonuç verdiği söylenebilir ancak güven aralığı kesişimleri incelendiğinde bu farkın anlamlı olmadığı görülmektedir.
- Hücre sayısı (HS) faktörünün algoritmalar performansları üzerinde etkisi bulunmaktadır. Hücre sayısı arttıkça algoritma performansları düşmektedir.
- Hücre parti sayısı (HPS) faktörünün algoritma performansları üzerinde anlamlı bir etkisi bulunmamaktadır. Çözüm kalitesi hücrelere çizelgelenmek üzere atanan partilerin sayılarından bağımsızdır.
- Hücre makine sayısı (HMS) faktörünün algoritma performansları üzerinde anlamlı bir etkisi bulunmamaktadır. Çözüm kalitesi hücrelere çizelgelenmek üzere atanan partilerin sayılarından bağımsızdır.
- Toplam çalışan sayısı (TÇS) faktörünün algoritma performansları üzerinde anlamlı bir etkisi bulunmamaktadır. Çözüm kalitesi hücrelere çizelgelenmek üzere atanan partilerin sayılarından bağımsızdır.

Büyük boyutlu problemler üretilirken her problem sınıfı için 10 farklı çalışan havuzu kullanılarak 10 farklı örnek oluşturulmuştur. Bu sayede çalışan havuzlarının çözüm kalitesi üzerindeki etkileri incelenmiştir.

Aşağıda çalışan havuzları için elde edilen sonuçların özetine yer verilmektedir.

- En düşük maliyet çoğunluğunu yetenek seviyesi düşük çalışanların oluşturduğu dokuzuncu havuz kullanıldığında elde edilmektedir.
- Orta maliyet değerlerine yetenek seviyesi orta çalışanlardan oluşan yedinci havuz kullanıldığında ulaşılmaktadır.

- En yüksek maliyet değerlerine yetenek seviyesi yüksek çalışanlardan oluşan dördüncü havuz kullanıldığında ulaşılmaktadır.
- Parça başı maliyet değerlerine bakıldığında da en iyi sonuçların dokuzuncu havuz kullanıldığında elde edildiği görülmektedir.
- Kullanılan çalışan yetenek seviye katsayıları dikkate alındığında en iyi sonuçlara yetenek seviyesi düşük çalışanlar (66%) ve yetenek seviyesi orta çalışanlardan (33%) oluşan dokuzuncu havuz ile ulaşılmaktadır. Bu durumun nedeni yetenek seviyesi yüksek çalışanların hücre çevrim sürelerini maliyet katsayıları oranınca yeterince düşürememesidir. Böylece hücre çevrim sürelerinde büyük oranlarda değişim görülmeyen sistemlerde yetenekli çalışanlar için yüksek maliyetlere katlanılmasına gerek yoktur sonucu çıkarılmıştır.

Uygulanan yöntem bilim sayesinde ÇKK HÜS' de çalışan verimlilik artışları meydana gelmiştir. Çalışan verimliliğin ölçülmesi için geliştirilen ÇDKO oranı uygulanan yöntem bilim sonrasında yaklaşık 20% artış göstermiştir. Verimlilik artışının temel nedeni çalışanların operasyonlara dahil olma sürelerinin yayılma zamanı içerisindeki payının artmasıdır. Bu artış incelendiğinde sisteme ne kadar çok parti salınımı gerçekleştirilirse o derece ÇDKO oranının arttığı görülmektedir. Bu tez kapsamında ele alınan gerçek üretim tesisinde toplamda dokuz çalışan bulunduğu ve her çalışanın yetenek seviyesinin orta olduğu varsayımıyla hareket edildiği için yedinci çalışan havuzundan yararlanılmıştır. Eğer çalışanların yetenek seviyeleri yüksek olarak varsayılırsa bu durumda verimlilik artış oranı 32% olmaktadır. Ancak toplam çalışan maliyetinde artışlar meydana gelmektedir. Sonuç olarak her çalışan havuzu varsayımı için verimlilikte artış meydana geldiği açıktır. Bu sonuçlar üzerinden önerilen yöntem bilim sayesinde uygulama gerçekleştirilen işletmede verimlilik artışı sağlandığı gözlemlenmiştir.

Bu tez kapsamında bir yöntem bilim ve iki farklı probleme yer verilmiştir. Her iki problem ele alınan üretim sistemi bakımından özgündür ve daha önce akademik literatürde ele alınmamıştır. Özellikle PÇÇA problemi hem parti çizelgelemeyi hem de çalışan atamayı hücre çevrim sürelerini göz önünde bulundurarak ele aldığı için gelecek çalışmalarda üzerinde durulması gereken bir problem türüdür. Bu nedenle aşağıda gelecek çalışmalar için önerilerde bulunmaktadır.

Bu tez içerisinde çalışanların atanma kararları ele alınmıştır. Atanma kararları parti çizelgeleme kararları ile birlikte verildiği için çalışan çizelgeleme de eş zamanlı olarak gerçekleştirilmektedir. PÇÇA problemi için geliştirilen matematiksel model de çalışanların işyüklerine yer verilmektedir ancak işyükleri arasındaki dengesizlikler göz önüne alınmamıştır. Bu durum dengesiz yüklemelere neden olabileceği ve uzun vadede performansları etkileyiciği için gelecek çalışmalarda gerek kısıtlar üzerinden gerekse de çok amaçlı bir matematiksel modelleme üzerinden bu durum ele alınabilir.

Bu tez kapsamında çalışanlar için hem işlem sürelerinin hesabında hem de maliyet hesabında çizelge 5.5 içerisinde yer verilen katsayılar kullanılmıştır. Bu değerler üzerinden gerçekleştirilen hesaplamalar sayesinde kullanılan parametrelerin önemli olduğu anlaşılmıştır. Bu nedenle gelecek çalışmalarda bu katsayılar bulanık mantık çerçevesinde ele alınarak duyarlılık analizleri gerçekleştirilebilir.

Parti salınım problemi için geliştirilen matematiksel model periyodik salınımına olanak sağlamaktadır. Gelecek çalışmalarda farklı salınım politikaları (melez salınım) benzer bir yöntemle içerisinde uygulanarak performansı değerlendirilebilir.

Yukarıda bahsedildiği üzere PÇÇA matematiksel modelinde çalışan uygunluk kısıtları ve işyüklerine yer verilmiştir. Ancak çalışanların hücreler arası hareketine yer verilmemiştir. Hücre çevrim süreleri atanan çalışanlara göre değişim göstermekte ve çalışanlar hücreler arası hareket edebilmektedirler. Bu durum bazı çalışanların yardımcı işgören olarak değerlendirilmesine olanak sağlamaktadır. Eğer bazı çalışanlar sadece yardımcı işgören olarak değerlendirilirlerse ve geri kalan çalışanlar asli görevlerine odaklanırlarsa bu durumda ÇDKO oranlarında artışlar mümkündür. Yardımcı işgören kavramı özellikle montaj hatlarında kullanılmaktadır (Çevikcan ve Durmuşoğlu, 2011) ve bu kavram ÇKK HÜS için uyarlanabilir.

Bu tez kapsamında PÇÇA problemine çözüm getirebilmek adına GA beş farklı öncelik kuralı ile birlikte uygulanmıştır. Gelecek çalışmalarda farklı meta-sezgiseller yöntemler uygulanabileceği gibi problem spesifik sezgiseller de geliştirilebilir.

Gelecek çalışmalarda ele alınabilecek bir diğer nokta ise parti büyüklükleridir. Bu tez kapsamında verilen sipariş sayıları aynı zamanda parti büyüklüklerine eşit kabul edilmiştir. Hem PS hem de PÇÇA problemleri için geliştirilen matematiksel modellerde bu varsayımı göz önünde bulundurulmuştur. Ancak parti büyüklükleriyle

alıřan atama ve parti izelgeleme kararları eř zamanlı olarak deęerlendirilirse gerek maliyetlerde gerekse de DKO oranında azalıřlar meydana gelebilecektir. Bu nedenle gelecek alıřmalarda bu konuya odaklanılmasında fayda bulunmaktadır.

KAYNAKLAR

- Adamu, M. O. & Adewumi, A.** (2015). Comparing the performance of different meta-heuristics for unweighted parallel machine scheduling. *South African Journal of Industrial Engineering*, 26(2), 143-157.
- Adewumi, A. O. & Ali, M. M.** (2010). A multi-level genetic algorithm for a multi-stage space allocation problem. *Mathematical and Computer Modelling*, 51(1), 109-126.
- Adewumi, A. O., Sawyerr, B. A. & Montaz Ali, M.** (2009). A heuristic solution to the university timetabling problem. *Engineering Computations*, 26(8), 972-984.
- Ağlan, C. & Durmusoğlu, M. B.** (2015). Lot-splitting approach of a hybrid manufacturing system under CONWIP production control: a mathematical model. *International Journal of Production Research*, 53(5), 1561-1583.
- Altay, A.** (2015). *Genetik algoritma ve bir uygulama* (Doctoral dissertation, Fen Bilimleri Enstitüsü).
- Ashby, J. R. & Uzsoy, R.** (1995). Scheduling and order release in a single-stage production system. *Journal of manufacturing Systems*, 14(4), 290-306.
- Askin, R. G. & Huang, Y.** (2001). Forming effective worker teams for cellular manufacturing. *International Journal of Production Research*, 39(11), 2431-2451.
- Azadeh, A., Rezaei-Malek, M., Evazabadian, F. & Sheikhalishahi, M.** (2014). Improved design of CMS by considering operators decision-making styles. *International Journal of Production Research*, 53(11), 3276–3287.
- Balaji, A. N. & Porselvi, S.** (2014). Artificial Immune System Algorithm and Simulated Annealing Algorithm for Scheduling Batches of Parts based on Job Availability Model in a Multi-cell Flexible Manufacturing System. *Procedia Engineering*, 97, 1524-1533.
- Batsis, A. E. & Tatsiopoulos, I. P.** (1993, September). A Methodology and Simulation of Workload Control in Cellular Manufacturing. In *Proceedings of the IFIP TC5/WG5. 7 Fifth International Conference on Advances in Production Management Systems* (pp. 143-151). North-Holland Publishing Co..
- Black, J. T. & Hunter, S. L.** (2003). *Lean manufacturing systems and cell design*. Society of Manufacturing Engineers.
- Boe, W. J. & Cheng, C. H.** (1991). A close neighbour algorithm for designing cellular manufacturing systems. *The International Journal of Production Research*, 29(10), 2097-2116.

- Celano, G., Costa, A. & Fichera, S.** (2008). Scheduling of unrelated parallel manufacturing cells with limited human resources. *International Journal of Production Research*, 46(2), 405-427.
- Cesani, V. I. & Steudel, H. J.** (2005). A study of labor assignment flexibility in cellular manufacturing systems. *Computers & Industrial Engineering*, 48(3), 571-591.
- Chandrasekharan, M. P. & Rajagopalan, R.** (1986). An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research*, 24(2), 451-463.
- Chen, Y. C. & Lee, C. E.** (2001). A bottleneck-based group scheduling procedure for job-shop cells. *Journal of the Chinese Institute of Industrial Engineers*, 18(5), 1-12.
- Coley, D. A.** (1999). *An introduction to genetic algorithms for scientists and engineers*. World Scientific Publishing Co Inc.
- Cortes, P., Larrañeta, J., Onieva, L., García, J. M. & Caraballo, M. S.** (2001). Genetic algorithm for planning cable telecommunication networks. *Applied Soft Computing*, 1(1), 21-33.
- Costa, A., Cappadonna, F. A. & Fichera, S.** (2014). Joint optimization of a flow-shop group scheduling with sequence dependent set-up times and skilled workforce assignment. *International Journal of Production Research*, 52(9), 2696-2728.
- Costa, A., Fichera, S. & Cappadonna F.** (2013) A hybrid genetic algorithm for job sequencing and worker allocation in parallel unrelated machines with sequence-dependent setup times. *Int J Adv Manuf Tech*, 69(9-12), 2799-2817.
- Çevikcan, E. & Durmusoğlu, M. B.** (2011). Minimising utility work and utility worker transfers for a mixed-model assembly line. *International Journal of Production Research*, 49(24), 7293-7314.
- Daniels, R. L., Hoopes, B. J. & Mazzola, J. B.** (1996). Scheduling parallel manufacturing cells with resource flexibility. *Management science*, 42(9), 1260-1276.
- Das, S. R. & Canel, C.** (2005). An algorithm for scheduling batches of parts in a multi-cell flexible manufacturing system. *International Journal of Production Economics*, 97(3), 247-262.
- Durmuşoğlu, M. B. & Satoğlu, Ş. I.** (2011). Axiomatic design of hybrid manufacturing systems in erratic demand conditions. *International Journal of Production Research*, 49(17), 5231-5261.
- Edis, E. B., Oguz, C. & Ozkarahan, I.** (2013). Parallel machine scheduling with additional resources: Notation, classification, models and solution methods. *European Journal of Operational Research*, 230(3), 449-463.
- Egilmez, G., Erenay, B. & Süer, G. A.** (2014). Stochastic skill-based manpower allocation in a cellular manufacturing system. *Journal of Manufacturing Systems*, 33(4), 578-588.

- Fan, J., Cao, M. & Feng, D.** (2010, June). Multi-objective dual resource-constrained model for cell formation problem. In *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on* (pp. 1031-1036). IEEE.
- Feng, H., Da, W., Xi, L., Pan, E. & Xia, T.** (2017). Solving the integrated cell formation and worker assignment problem using particle swarm optimization and linear programming. *Computers & Industrial Engineering*.
- Fernandes, N. O. & Carmo-Silva, S.** (2011). Workload control under continuous order release. *International Journal of Production Economics*, 131(1), 257-262.
- Fernandes, N. O., Thürer, M., Silva, C. & Carmo-Silva, S.** (2016). Improving workload control order release: incorporating a starvation avoidance trigger into continuous release. *International Journal of Production Economics*.
- Germes, R. & Riezebos, J.** (2010). Workload balancing capability of pull systems in MTO production. *International Journal of Production Research*, 48(8), 2345-2360.
- Gupta, J. N. D. & Schaller, J. E.** (2005). Minimizing flow time in a flow-line manufacturing cell with family setup times. *Journal of the Operational Research Society*, 57(2), 163–176.
- Hachicha, W., Masmoudi, F. & Haddar, M.** (2007). An improvement of a cellular manufacturing system design using simulation analysis. *International Journal of Simulation Modelling*, 6(4), 193-205.
- Ham, I., Hitomi, K. & Yoshida, T.** (1985). *Basic Principles of Group Technology*. Springer Netherlands.
- Hamza, S. A. & Adesta, E. Y. T.** (2013). Similarity coefficient measures applied to integrate feasibility assessment and the design of cellular manufacturing systems. *Australian Journal of Basic and Applied Sciences*, 7(6), 257-266.
- Holland, J. H.** (1992). Genetic algorithms. *Scientific american*, 267(1), 66-72.
- Hottenstein, M. P. & Bowman, S. A.** (1998). Cross-training and worker flexibility: A review of DRC system research. *The Journal of High Technology Management Research*, 9(2), 157-174.
- Hyer, H. & Wemmerlov, U.** (2002). *Reorganizing the Factory Competing Through Cellular Manufacturing*, Productivity Press.
- Jensen, J. B.** (2000). The impact of resource flexibility and staffing decisions on cellular and departmental shop performance. *European Journal of Operational Research*, 127(2), 279-296.
- Kanet, J. J.** (1988). Load-limited order release in job shop scheduling systems. *Journal of operations management*, 7(3-4), 44-58.
- Kellerer, H. & Strusevich, V.A.** (2008). Scheduling parallel dedicated machines with the speeding-up resource. *Naval Research Logistics*, 55(5), 377-389.

- King, J. R.** (1980). Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. *International Journal of Production Research*, 18(2), 213-232.
- Kolbe, K.** (1993). *Pure Instinct*. Random House: New York.
- Kolisch, R.** (1996). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, 90(2), 320-333.
- Kuo, Y. & Liu, C. C.** (2017). Operator Assignment in a Labor-Intensive Manufacturing Cell Considering Inter-Cell Manpower Transfer. *Computers & Industrial Engineering*.
- Land, M.** (2006). Parameters and sensitivity in workload control. *International Journal of Production Economics*, 104(2), 625–638.
- Land, M. J. & Gaalman, G. J.** (1998). The performance of workload control concepts in job shops: improving the release method. *International Journal of Production Economics*, 56, 347-364.
- Li, Y., Li, X. & Gupta, J. N.** (2015). Solving the multi-objective flowline manufacturing cell scheduling problem by hybrid harmony search. *Expert Systems with Applications*, 42(3), 1409-1417.
- Li, X. & Gao, L.** (2016). An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. *International Journal of Production Economics*, 174, 93-110.
- Liao, C. J.** (1992). Optimal control of jobs for production systems. *Computers & industrial engineering*, 22(2), 163-169.
- Mahmoodi, F., Dooley, K. J. & Starr, P. J.** (1990). An evaluation of order releasing and due date assignment heuristics in a cellular manufacturing system. *Journal of Operations Management*, 9(4), 548-573.
- Man, K. F., Tang, K. S. & Kwong, S.** (2012). *Genetic algorithms: Concepts and designs*. Springer Science & Business Media.
- Miller, B. L. & Goldberg, D. E.** (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3), 193-212.
- Miltenburg, J.** (2001). One-piece flow manufacturing on U-shaped production lines: a tutorial, *IIE transactions*, 33(4), 303-321.
- Mönch, L.** (2005, December). Simulation-based assessment of order release strategies for a distributed shifting bottleneck heuristic. In *Proceedings of the 37th conference on Winter simulation* (pp. 2186-2193). Winter Simulation Conference.
- Naso, D., Surico, M., Turchiano, B. & Kaymak, U.** (2007). Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete. *European Journal of Operational Research*, 177(3), 2069-2099.
- Norman, B. A., Tharmmaphornphilas, W., Needy, K. L., Bidanda, B. & Warner, R. C.** (2002). Worker assignment in cellular manufacturing

- considering technical and human skills. *International Journal of Production Research*, 40(6), 1479-1492.
- Park, M. W. & Kim, Y. D.** (1998). A systematic procedure for setting parameters in simulated annealing algorithms. *Computers & Operations Research*, 25(3), 207-217.
- Qi, C., Sivakumar, A. I. & Gershwin, S. B.** (2009). An efficient new job release control methodology. *International Journal of Production Research*, 47(3), 703-731.
- Quagliarella, D.** (1998). *Genetic algorithms and evolution strategy in engineering and computer science: recent advances and industrial applications*. John Wiley & Son Ltd.
- Quintana, R.** (2002). Recursive linear control of order release to manufacturing cells with random yield. *Iie Transactions*, 34(5), 489-500.
- Rajesh, K. D., Krishna, M. M., Ali, M. A. & Chalapathi, P. V.** (2017). A Modified Hybrid Similarity Coefficient Based Method for Solving the Cell Formation Problem in Cellular Manufacturing System. *Materials Today: Proceedings*, 4(2), 1469-1477.
- Ramesh, S., Kannan, S. & Baskar, S.** (2012). Application of modified NSGA-II algorithm to multi-objective reactive power planning. *Applied Soft Computing*, 12(2), 741-753.
- Sabuncuoglu, I. & Karapinar, H. Y.** (1999). Analysis of order review/release problems in production systems. *International Journal of Production Economics*, 62(3), 259-279.
- Satoğlu, Ş. I., Durmuşoğlu, M. B. & Ertay, T.** (2010). A mathematical model and a heuristic approach for design of the hybrid manufacturing systems to facilitate one-piece flow. *International Journal of Production Research*, 48(17), 5195-5220.
- Satoğlu, Ş. I. & Suresh, N. C.** (2009). A goal-programming approach for design of hybrid cellular manufacturing systems in dual resource constrained environments. *Computers & industrial engineering*, 56(2), 560-575.
- Schaller, J. E., Gupta, J. N. & Vakharia, A. J.** (2000). Scheduling a flowline manufacturing cell with sequence dependent family setup times. *European Journal of Operational Research*, 125(2), 324-339.
- Solimanpur, M. & Elmi, A.** (2011). A tabu search approach for group scheduling in buffer-constrained flow shop cells. *International Journal of Computer Integrated Manufacturing*, 24(3), 257-268.
- Süer, G. A. & Alhawari, O.** (2011). Operator assignment decisions in a highly dynamic cellular environment. *Operations Management Research and Cellular Manufacturing Systems: Innovative Methods and Approaches: Innovative Methods and Approaches*, 258.
- Süer, G. A. & Dagli, C.** (2005). Intra-cell manpower transfers and cell loading in labor-intensive manufacturing cells. *Computers & Industrial Engineering*, 48(3), 643-655.

- Süer, G. A. & Tummaluri, R. R.** (2008). Multi-period operator assignment considering skills, learning and forgetting in labour-intensive cells. *International Journal of Production Research*, 46(2), 469-493.
- Thürer, M., Land, M. J., Stevenson, M., Fredendall, L. D. & Godinho Filho, M.** (2015). Concerning Workload Control and Order Release: The Pre-Shop Pool Sequencing Decision. *Production and Operations Management*, 24(7), 1179-1192.
- Thürer, M., Silva, C., Stevenson, M. & Land, M.** (2014). Controlled order release: a performance assessment in job shops with sequence-dependent set-up times. *Production Planning & Control*, 25(7), 603-615.
- Ünal, M., Ak, A., Topuz, V. & Erdal, H.** (2012). *Optimization of PID controllers using ant colony and genetic algorithms* (Vol. 449). Springer.
- Venkataramanaiah, S.** (2008). Scheduling in cellular manufacturing systems: an heuristic approach. *International Journal of Production Research*, 46(2), 429-449.
- Won, Y. & Currie, K.R.** (2004). Efficient p-median mathematical programming approaches to machine-part grouping in group technology manufacturing, *Engineering Optimization*, 36(5), 555-573.
- Xu, J., Xu, X. & Xie, S. Q.** (2011). Recent developments in Dual Resource Constrained (DRC) system research. *European journal of operational research*, 215(2), 309-318.
- Yin, Y. & Yasuda, K.** (2006). Similarity coefficient methods applied to the cell formation problem: A taxonomy and review. *International Journal of Production Economics*, 101(2), 329-352.
- Yin, Y. & Yasuda, K.** (2005). Similarity coefficient methods applied to the cell formation problem: a comparative investigation. *Computers & Industrial Engineering*, 48(3), 471-489.
- Yılmaz, Ö. F., Çevikcan, E. & Durmuşoğlu, M. B.** (2016). Scheduling batches in multi hybrid cell manufacturing system considering worker resources: A case study from pipeline industry. *Advances in Production Engineering & Management*, 11(3), 192.
- Yılmaz, Ö. F. & Durmuşoğlu, M. B.** (2017). An Integrated Methodology for Order Release and Scheduling in Hybrid Manufacturing Systems: Considering Worker Assignment and Utility Workers. *Handbook of Research on Applied Optimization Methodologies in Manufacturing Systems* (pp. 125-161). IGI Global.

EKLER

EK A: Matlab ve GAMS Kodlar

EK A

```
function [obj,obj1,lb,FLT,cycmin,cycmax,time] =  
genetic(population_size,pcross,pmutation,objevaluation,processing,q,walkingtime,at  
amalar,workerskill,workerskillcost)  
iteration_number=round(objevaluation/population_size);  
numberofbatch=size(processing,1);  
numberofcell=max(atamalar);  
numberofworker=size(workerskill,2);  
numberofmachine=size(processing,2)/2;  
for i=1:numberofbatch  
for k=1:numberofmachine  
toplak(i,k)=processing(i,2*k-1)+processing(i,2*k);  
end  
end  
toplak1=toplak';  
cycmin=max(toplak1);  
for i=1:numberofbatch  
hala(i)=0;  
for k=1:numberofmachine  
hala(i)=hala(i)+processing(i,2*k-1)+walkingtime(i);  
end  
end  
for i=1:numberofbatch  
cycmax(i)=max(hala(i),max(toplak(i,:)));  
end  
for t=1:population_size  
chromosome1(:,t)=zeros(2,numberofbatch);  
chromosome1(1,:,t)=randperm(numberofbatch);  
for i=1:numberofbatch  
maxatacalisan(i,t)=ceil(cycmax(i)/cycmin(i));  
atacalisan(i,t)=randi(maxatacalisan(i,t));  
end  
for i=1:numberofbatch  
chromosome1(2,i,t)=atacalisan(chromosome1(1,i,t),t);  
end  
chromosome2(:,t)=zeros(2,sum(maxatacalisan(:,t)));  
k=0;  
for i=1:numberofbatch  
chromosome2(1,k+1:k+chromosome1(2,i,t),t)=randperm(numberofworker,chromoso  
me1(2,i,t));  
x=zeros(1,chromosome1(2,i,t));  
for j=1:chromosome1(2,i,t)  
x(1,j)=randi(ceil(numberofmachine/chromosome1(2,i,t))-1);  
end  
if(sum(x)<=numberofmachine)  
tt=find(x==min(x),1);  
x(tt)=x(tt)+(numberofmachine-sum(x));  
end  
tt1=find(x==min(x),1);
```

```

    tt2=find(x==max(x),1);
    if (x(tt2)-x(tt1))>=2)
        tt3=randi(x(tt2)-x(tt1));
        x(tt1)=x(tt1)+tt3;
        x(tt2)=x(tt2)-tt3;
    end
    chromosome2(2,k+1:k+chromosome1(2,i,t),t)=x;
    k=maxatacalisan(chromosome1(1,i,t),t)+k;
end
k=0;
for i=1:numberofbatch
    chromosome3(1,k+1:k+numberofmachine,t)=randperm(numberofmachine);
    k=numberofmachine+k;
end
population1(:,t)=chromosome1(:,t);
population2(:,t)=chromosome2(:,t);
population3(:,t)=chromosome3(:,t);
end
[p,s,c,cyc,chromosome1,chromosome2,chromosome3,objective,FLT] =
objectivefunction(population1,population2,population3,numberofbatch,numberofcell
,maxatacalisan,numberofmachine,population_size,workerskill,processing,q,walkingti
me,cycmin,atamalar,workerskillcost);
best_obj=min(objective(1,:));
best_obj1=objective(2,find(objective(1,:)==best_obj,1));
best_lb=objective(3,find(objective(1,:)==best_obj,1));
iteration=1;
obj(iteration)=best_obj;
obj1(iteration)=best_obj1;
lb(iteration)=best_lb;
asda=clock;
while (iteration<=iteration_number);
    m1=population1;
    m2=population2;
    m3=population3;
    abc=sort(objective(1,:));
    bir=find(objective(1,:)==abc(1,1),1);
    birobj=abc(1,1);
    birobj1=objective(2,bir);
    birlb1=objective(3,bir);
    iki=find(objective(1,:)==abc(1,2),1);
    ikiobj=abc(1,2);
    ikiobj1=objective(2,iki);
    ikilb1=objective(3,iki);
    [mid_population1,mid_population2,mid_population3 ] = selection(
    population1,population2,population3,objective,population_size );
    [ mid_population1,mid_population2,mid_population3 ] = croosover(
    mid_population1,mid_population2,mid_population3,pcross,maxatacalisan,numberof
    machine,population_size,numberofbatch );
end

```

```

[ mid_population1,mid_population2,mid_population3 ] = mutation(
pmutation,mid_population1,mid_population2,mid_population3,population_size,num
berofbatch,maxatacalisan,numberofmachine);
population1=mid_population1;
population2=mid_population2;
population3=mid_population3;
[p,s,c,cyc,chromosome1,chromosome2,chromosome3,objective,FLT] =
objectivefunction(population1,population2,population3,numberofbatch,numberofcell
,maxatacalisan,numberofmachine,population_size,workerskill,processing,q,walkingti
me,cycmin,atamalar,workerskillcost);
abc1=sort(objective(1,:));
uc=find(objective(1,:)==abc1(1,population_size),1);
dort=find(objective(1,:)==abc1(1,population_size-1),1);
if(objective(1,uc)>biobj)
population1(:,uc)=m1(:,bir);
population2(:,uc)=m2(:,bir);
population3(:,uc)=m3(:,bir);
objective(1,uc)=biobj;
objective(2,uc)=biobj1;
objective(3,uc)=birlb1;
end
if(objective(1,dort)>ikiobj)
population1(:,dort)=m1(:,iki);
population2(:,dort)=m2(:,iki);
population3(:,dort)=m3(:,iki);
objective(1,dort)=ikiobj;
objective(2,dort)=ikiobj1;
objective(3,dort)=ikilb1;
end
best_obj=min(objective(1,:));
best_obj1=objective(2,find(objective(1,:)==best_obj,1));
best_lb=objective(3,find(objective(1,:)==best_obj,1));
iteration=iteration+1;
obj(iteration)=best_obj;
obj1(iteration)=best_obj1;
lb(iteration)=best_lb;
end
time=etime(clock,asda);
end

```

```

function [ mid_population1,mid_population2,mid_population3 ] = croosover(
mid_population1,mid_population2,mid_population3,pcross,maxatacalisan,numberof
machine,population_size,numberofbatch )
pairs=randperm(population_size);
for i=1:population_size/2
if (unifrnd(0,1)<=pcross)
cpoint=unidrnd(length(mid_population1(1,:,1))-1);
abc1=mid_population1(:,1:cpoint,pairs(2*i-1));
abc7=mid_population1(:,1:cpoint,pairs(2*i));
bul=zeros(1,cpoint);

```



```

    bul1=zeros(1,cpoint);
    k=0;
    k1=0;
    for j=1:cpoint
        k=k+maxatacalisan(mid_population1(1,j,pairs(2*i-1)),pairs(2*i-1));
        k1=k1+maxatacalisan(mid_population1(1,j,pairs(2*i)),pairs(2*i));

        bul(1,j)=find(mid_population1(1,,:,pairs(2*i))==abc1(1,j));
        bul1(1,j)=find(mid_population1(1,,:,pairs(2*i-1))==abc7(1,j));
    end
    abc2=mid_population2(:,1:k,pairs(2*i-1));
    abc8=mid_population2(:,1:k1,pairs(2*i));
    abc3=mid_population3(1,1:numberofmachine*cpoint,pairs(2*i-1));
    abc9=mid_population3(1,1:numberofmachine*cpoint,pairs(2*i));
    theset=1:length(mid_population1(1,:,1));
    idx=ismember(theset,bul);
    idx1=ismember(theset,bul1);
    missing=theset(~idx);
    missing1=theset(~idx1);
    abc4=mid_population1(:,missing,pairs(2*i));
    abc10=mid_population1(:,missing1,pairs(2*i-1));
    az(1,1)=maxatacalisan(mid_population1(1,1,pairs(2*i)),pairs(2*i));
    azz(1,1)=maxatacalisan(mid_population1(1,1,pairs(2*i-1)),pairs(2*i-1));
    az1(1,1)=az(1,1);
    azz1(1,1)=azz(1,1);
    for k=2:numberofbatch
        az(1,k)=az(1,k-1)+maxatacalisan(mid_population1(1,k,pairs(2*i)),pairs(2*i));
        azz(1,k)=azz(1,k-1)+maxatacalisan(mid_population1(1,k,pairs(2*i-
1)),pairs(2*i-1));
        az1(1,k)=maxatacalisan(mid_population1(1,k,pairs(2*i)),pairs(2*i));
        azz1(1,k)=maxatacalisan(mid_population1(1,k,pairs(2*i-1)),pairs(2*i-1));
    end
    as=zeros(1,length(abc4(1,:)));
    as1=zeros(1,length(abc10(1,:)));
    abc5=zeros(2,length(mid_population2(1,,:,pairs(2*i)))-length(abc2(1,:)));
    abc11=zeros(2,length(mid_population2(1,,:,pairs(2*i-1)))-length(abc8(1,:)));
    abc6=zeros(1,numberofmachine*length(abc4(1,:)));
    abc12=zeros(1,numberofmachine*length(abc10(1,:)));
    as(1,1)=az1(1,find(mid_population1(1,,:,pairs(2*i))==abc4(1,1)));
    as1(1,1)=azz1(1,find(mid_population1(1,,:,pairs(2*i-1))==abc10(1,1)));

    abc5(:,1:as(1,1))=mid_population2(:,az(1,find(mid_population1(1,,:,pairs(2*i))==abc
4(1,1)))-
    az1(1,find(mid_population1(1,,:,pairs(2*i))==abc4(1,1)))+1:az(1,find(mid_populatio
n1(1,,:,pairs(2*i))==abc4(1,1))),pairs(2*i));

    abc11(:,1:as1(1,1))=mid_population2(:,azz(1,find(mid_population1(1,,:,pairs(2*i-
1))==abc10(1,1)))-azz1(1,find(mid_population1(1,,:,pairs(2*i-
1))==abc10(1,1)))+1:azz(1,find(mid_population1(1,,:,pairs(2*i-
1))==abc10(1,1))),pairs(2*i-1));

```

```

        abc6(1,1:numberofmachine)=
mid_population3(1,(find(mid_population1(1,.,pairs(2*i))==abc4(1,1))-
1)*numberofmachine+1:(find(mid_population1(1,.,pairs(2*i))==abc4(1,1)))*number
ofmachine,pairs(2*i));
        abc12(1,1:numberofmachine)=
mid_population3(1,(find(mid_population1(1,.,pairs(2*i-1))==abc10(1,1))-
1)*numberofmachine+1:(find(mid_population1(1,.,pairs(2*i-
1))==abc10(1,1)))*numberofmachine,pairs(2*i-1));
        for k=2:length(abc4(1,:));
            as(1,k)=as(1,k-1)+az1(1,find(mid_population1(1,.,pairs(2*i))==abc4(1,k)));
            abc5(:,as(1,k-
1)+1:as(1,k))=mid_population2(:,az(1,find(mid_population1(1,.,pairs(2*i))==abc4(1,
k))))-
az1(1,find(mid_population1(1,.,pairs(2*i))==abc4(1,k)))+1:az(1,find(mid_populatio
n1(1,.,pairs(2*i))==abc4(1,k))),pairs(2*i));
            abc6(1,(k-1)*numberofmachine+1:k*numberofmachine)=
mid_population3(1,(find(mid_population1(1,.,pairs(2*i))==abc4(1,k))-
1)*numberofmachine+1:(find(mid_population1(1,.,pairs(2*i))==abc4(1,k)))*number
ofmachine,pairs(2*i));
        end
        for k=2:length(abc10(1,:));
            as1(1,k)=as1(1,k-1)+azz1(1,find(mid_population1(1,.,pairs(2*i-
1))==abc10(1,k)));
            abc11(:,as1(1,k-
1)+1:as1(1,k))=mid_population2(:,azz(1,find(mid_population1(1,.,pairs(2*i-
1))==abc10(1,k)))-azz1(1,find(mid_population1(1,.,pairs(2*i-
1))==abc10(1,k)))+1:azz(1,find(mid_population1(1,.,pairs(2*i-
1))==abc10(1,k))),pairs(2*i-1));
            abc12(1,(k-1)*numberofmachine+1:k*numberofmachine)=
mid_population3(1,(find(mid_population1(1,.,pairs(2*i-1))==abc10(1,k))-
1)*numberofmachine+1:(find(mid_population1(1,.,pairs(2*i-
1))==abc10(1,k)))*numberofmachine,pairs(2*i-1));
        end
        mid_population1(:,pairs(2*i-1))=[abc1 abc4];
        mid_population2(:,pairs(2*i-1))=[abc2 abc5];
        mid_population3(:,pairs(2*i-1))=[abc3 abc6];
        mid_population1(:,pairs(2*i))=[abc7 abc10];
        mid_population2(:,pairs(2*i))=[abc8 abc11];
        mid_population3(:,pairs(2*i))=[abc9 abc12];
    end
end
end

```

```

function [ mid_population1,mid_population2,mid_population3 ] = mutation(
pmutation,mid_population1,mid_population2,mid_population3,population_size,num
berofbatch,maxatacalisan,numberofmachine)
for j=1:population_size;
    pairs=randperm(numberofbatch-1)+1;
    for i=1:(numberofbatch/2)-1;
        az(1,1)=maxatacalisan(mid_population1(1,1,j),j);
    end
end
end

```

```

    for k=2:numberofbatch;
    az(1,k)=az(1,k-1)+maxatacalisan(mid_population1(1,k,j),j);
    end
    bak=randperm(population_size,1);
    bak1=randperm(population_size,1);
    as(1,1)=maxatacalisan(mid_population1(1,1,bak),bak);
    as1(1,1)=maxatacalisan(mid_population1(1,1,bak1),bak1);
    for k=2:numberofbatch;
    as(1,k)=as(1,k-1)+maxatacalisan(mid_population1(1,k,bak),bak);
    as1(1,k)=as1(1,k-1)+maxatacalisan(mid_population1(1,k,bak1),bak1);
    end
    if (unifrnd(0,1)<=pmutation);
    ata=[pairs(2*i-1) pairs(2*i)];
    ata1=sort(ata);
    a1=mid_population1(:,ata1(1),j);
    a2=mid_population1(:,ata1(2),j);
    a3=mid_population2(:,1:az(1,ata1(1))-1,j);
    a4=mid_population2(:,as(1,find(mid_population1(1,:,bak)==a1(1,1)))-
maxatacalisan(a1(1,1),bak)+1:as(1,find(mid_population1(1,:,bak)==a1(1,1))),bak);
    a5=mid_population2(:,az(1,ata1(1))+1:az(1,ata1(2))-1,j);
    a6=mid_population2(:,as1(1,find(mid_population1(1,:,bak1)==a2(1,1)))-
maxatacalisan(a2(1,1),bak1)+1:as1(1,find(mid_population1(1,:,bak1)==a2(1,1))),bak
1);
    a7=mid_population2(:,az(1,ata1(2))+1:az(1,numberofbatch),j);
    a1(2,1)=length(find(a4(1,:)>0));
    mid_population1(:,ata1(2),j)=a1;
    a2(2,1)=length(find(a6(1,:)>0));
    mid_population1(:,ata1(1),j)=a2;
    mid_population2(:,j)=[a3 a6 a5 a4 a7];
    a8=mid_population3(1,1:(ata1(1)-1)*numberofmachine,j);
    a9=mid_population3(1,(ata1(1)-
1)*numberofmachine+1:ata1(1)*numberofmachine,j);
    a10=mid_population3(1,ata1(1)*numberofmachine+1:(ata1(2)-
1)*numberofmachine,j);
    a11=mid_population3(1,(ata1(2)-
1)*numberofmachine+1:ata1(2)*numberofmachine,j);

    a12=mid_population3(1,ata1(2)*numberofmachine+1:numberofmachine*numberofb
atch,j);
    mid_population3(1,:,j)=[a8 a11 a10 a9 a12];
    end
end
end
end

function [p,s,c,cyc,chromosome1,chromosome2,chromosome3,objective,FLT] =
objectivefunction(population1,population2,population3,numberofbatch,numberofcell
,maxatacalisan,numberofmachine,population_size,workerskill,processing,q,walkingti
me,cycmin,atamalar,workerskillcost )
for k=1:numberofbatch

```

```

FLT(k)=sum(processing(k,:));
end
for z=1:population_size;
maxatacalisan(:,z)=maxatacalisan(:,1);
chromosome1(:,z)=population1(:,z);
chromosome2(:,z)=population2(:,z);
chromosome3(:,z)=population3(:,z);
asd(1,1,z)=chromosome2(2,1,z);
for i=2:length(chromosome2(1,:,z))
    asd(1,i,z)=asd(1,i-1,z)+chromosome2(2,i,z);
end
k=1;
for i=1:length(chromosome2(1,:,z))
    if(asd(1,i,z)>k*numberofmachine)
        k=k+1;
    end
    if (chromosome2(1,i,z)>0)
        cyc1(1,i,z)=(sum(processing(chromosome1(1,k,z),2*chromosome3(1,asd(1,i,z)-
chromosome2(2,i,z)+1:asd(1,i,z),z)-
1))/workerskill(1,chromosome2(1,i,z)))+chromosome2(2,i,z)*walkingtime(chromoso
me1(1,k,z));
    else
        cyc1(1,i,z)=0;
    end
end
end
azd(1,1,z)=maxatacalisan(chromosome1(1,1,z),z);
azd1(1,1,z)=azd(1,1,z);
for i=2:numberofbatch
    azd(1,i,z)=azd(1,i-1,z)+maxatacalisan(chromosome1(1,i,z),z);
    azd1(1,i,z)=maxatacalisan(chromosome1(1,i,z),z);
end
cyc(1,:,z)=zeros(1,numberofbatch);
for k=1:numberofbatch
    cyc(1,chromosome1(1,k,z),z)=max(max(cyc1(1,azd(1,k,z)-
azd1(1,k,z)+1:azd(1,k,z),z)),cycmin(chromosome1(1,k,z)));
end
p(:,z)=zeros(numberofbatch,numberofcell);
for k=1:numberofbatch
    p(k,atamalar(k),z)=(cyc(1,k,z)*(q(k)-1))+FLT(k);
end
p(:,z)=round(p(:,z));
kb(1,1,z)=maxatacalisan(chromosome1(1,1,z),z);
for i=2:numberofbatch
    kb(1,i,z)=kb(1,i-1,z)+maxatacalisan(chromosome1(1,i,z),z);
end
atananlar(:,z)=zeros(numberofbatch,max(maxatacalisan(:,z)));
atananlar(chromosome1(1,1,z),1:chromosome1(2,1,z),z)=chromosome2(1,1:chromos
ome1(2,1,z),z);
for i=2:numberofbatch

```

```

        atananlar(chromosome1(1,i,z),1:chromosome1(2,i,z),z)=chromosome2(1,kb(1,i-
1,z)+1:kb(1,i-1,z)+chromosome1(2,i,z),z);
    end
    kume1=chromosome1(1,:,z);
    ccc=kume1;
    kume2=zeros(2,length(chromosome1(1,:,z)));
    c(:,z)=zeros(numberofbatch,numberofcell);
    s(:,z)=zeros(numberofbatch,numberofcell);
    c(kume1(1),find(p(kume1(1),:,z)>0),z)=s(kume1(1),find(p(kume1(1),:,z)>0),z)+p(ku
me1(1),find(p(kume1(1),:,z)>0),z);
    klm=s(:,z);
    klm1=c(:,z);
    kume2(1,1)=kume1(1);
    kume2(2,1)=c(kume2(1,1),find(p(kume2(1,1),:,z)>0),z);
    kume2(3,1)=s(kume2(1,1),find(p(kume2(1,1),:,z)>0),z);
    kume3=kume2(:,1:find(kume2(2,:)>0));
    kume1(1)=[];
    for i=2:numberofbatch
        possible=sort(unique(klm1(find(klm1>=max(klm1(:,find(p(kume1(1),:,z)>0))))))'
        ;
        for k=1:length(possible)
            kume41=kume3(1,find(kume3(2,:)>possible(1,k)));

kume42=kume3(1,find(kume3(3,:)<=(possible(1,k)+p(kume1(1),find(p(kume1(1),:,z
)>0),z)))));
            kume4=intersect(kume41,kume42);
            cak=zeros(1,length(kume4));
            for j=1:length(kume4)
                j11=possible(1,k);
                j12=possible(1,k)+p(kume1(1),find(p(kume1(1),:,z)>0),z);
                j21=s(kume4(j),find(p(kume4(j),:,z)>0),z);

j22=s(kume4(j),find(p(kume4(j),:,z)>0),z)+p(kume4(j),find(p(kume4(j),:,z)>0),z);
                if (length(intersect(linspace(j11,j12,(j12-j11+1)),linspace(j21,j22,(j22-
j21+1))))>1)&&(sum(intersect(atananlar(kume1(1),:,z),atananlar(kume4(j),:,z)))>0)
                cak(1,j)=1;
            end
        end
        if (sum(cak(1,:))==0)
            s(kume1(1),find(p(kume1(1),:,z)>0),z)=possible(1,k);

c(kume1(1),find(p(kume1(1),:,z)>0),z)=s(kume1(1),find(p(kume1(1),:,z)>0),z)+p(ku
me1(1),find(p(kume1(1),:,z)>0),z);
            klm=s(:,z);
            klm1=c(:,z);
            kume2(1,i)=kume1(1);
            kume2(2,i)=c(kume2(1,i),find(p(kume2(1,i),:,z)>0),z);
            kume3=kume2(:,1:max(find(kume2(2,:)>0)));
            kume1(1)=[];
            break

```

```

        end
    end
end
for i=1:numberofbatch
    plower(i)=cycmin(i)*(q(i)-1)+FLT(i);
end
plowerbound=sum(plower)/numberofcell;
objective(1,z)=max(max(c(:,z)))*(sum(workerskillcost));
objective(2,z)=max(max(c(:,z)));
objective(3,z)=plowerbound*(sum(workerskillcost));
end
end

```

```

function [ mid_population1,mid_population2,mid_population3 ] = selection(
population1,population2,population3,objective,population_size )
objective(1,:)=1./objective(1,:);
sumobj=sum(objective(1,:));
probs=objective(1,:)/sumobj;
cprobs=probs;
for i=2:population_size
    cprobs(i)=cprobs(i-1)+probs(i);
end
rs=unifrnd(0,1,[population_size,1]);
mid_population1=population1;
mid_population2=population2;
mid_population3=population3;
for i=1:population_size
    idx=find(rs(i)<cprobs,1);
    mid_population1(:,i)=population1(:,idx);
    mid_population2(:,i)=population2(:,idx);
    mid_population3(:,i)=population3(:,idx);
end
end

```

```

GAMS
sets    i products /1*12/ ;
sets    k cells /1*3/ ;
sets    w numberofworkers /1*3/;
sets    m numberofmachines /1*4/;
sets    t timeindex /1*16673/;
        alias (i,j)
        alias (l,k)
scalar em  buyuk bir sayi /16673/ ;
parameters
t1(t)
$call GDXXRW data1.xlsx par=t1 rng=t1!a1:b16673 rdim=1 cdim=0
$GDXIN data1.gdx
$LOAD t1
$GDXIN
ak(i,k)

```

```

$call GDXXRW data1.xlsx par=ak rng=ak!o1:r13 rdim=1 cdim=1
$GDXIN data1.gdx
$LOAD ak
$GDXIN
manual(i,m)
$call GDXXRW data1.xlsx par=manual rng=manual!aj1:an13 rdim=1 cdim=1
$GDXIN data1.gdx
$LOAD manual
$GDXIN
FLT(i)
$call GDXXRW data1.xlsx par=FLT rng=FLT!v1:w12 rdim=1 cdim=0
$GDXIN data1.gdx
$LOAD FLT
$GDXIN
q(i)
$call GDXXRW data1.xlsx par=q rng=q!j1:k12 rdim=1 cdim=0
$GDXIN data1.gdx
$LOAD q
$GDXIN
cycmin(i)
$call GDXXRW data1.xlsx par=cycmin rng=cycmin!v1:w12 rdim=1 cdim=0
$GDXIN data1.gdx
$LOAD cycmin
$GDXIN
maxata(i)
$call GDXXRW data1.xlsx par=maxata rng=maxata!v1:w12 rdim=1 cdim=0
$GDXIN data1.gdx
$LOAD maxata
$GDXIN
walking(i)
$call GDXXRW data1.xlsx par=walking rng=walking!j1:k12 rdim=1 cdim=0
$GDXIN data1.gdx
$LOAD walking
$GDXIN
maliyet(w)
$call GDXXRW data1.xlsx par=maliyet rng=maliyet!c1:d3 rdim=1 cdim=0
$GDXIN data1.gdx
$LOAD maliyet
$GDXIN
variables
time(j)
p(i)
s(i)
c(i)
c1
cyc(i)
x(w,i,m)
x1(w,i)
b(i,j)
ka(i,j)

```

```

kb(i,j)
g(i,j)
t11(i,t)
amac;
binary variables
b(i,j)
ka(i,j)
kb(i,j)
x(w,i,m)
x1(w,i)
t11(i,t)
g(i,j);
equations
objective
kisit3(i,j)
kisit4(i)
kisit41(i)
kisit5(i)
kisit51(i)
kisit7(i)
kisit6(i,j)
kisit61(i,j)
kisit62(i,j)
kisit9(j)
kisit11(i,j)
kisit13(i,j)
kisit15(i)
kisit16(i,w)
kisit161(i,m)
kisit17(i)
kisit24(w,i,m)
kisit19(w,i,j)
kisit20(i)
objective..
kisit3(i,j)..
kisit4(i)..
kisit41(i)..
kisit5(i)..
kisit51(i)..
kisit7(i)..
kisit6(i,j)$ (ord(i) le 4 and ord(j) le 4 and ord(i) ne ord(j) )..      b(i,j)+b(j,i)=e=1;
kisit61(i,j)$ (ord(i) gt 8 and ord(j) gt 8 and ord(i) ne ord(j) )..      b(i,j)+b(j,i)=e=1;
kisit62(i,j)$ (ord(i) gt 4 and ord(i) le 8 and ord(j) gt 4 and ord(j) le 8 and ord(i) ne
ord(j) ).. b(i,j)+b(j,i)=e=1;
kisit9(j)..
kisit11(i,j)$ (ord(i) ne ord(j))..
em*ka(i,j);
kisit13(i,j)$ (ord(i) ne ord(j))..
kisit15(i)..
amac=e=c1*sum(w,maliyet(w));
c(i)=g=((c(j)+p(i))-em*(1-b(i,j)));
c(i)=g=p(i);
s(i)=e=c(i)-p(i);
s(i)=e=sum(t,ord(t)*t11(i,t));
sum(t,t11(i,t))=e=1;
c1=g=c(i);
time(j)=e=(c(j)-p(j));
time(j)-(c(i)-p(i))=l=
c(i)-time(j)=l=em*kb(i,j);
p(i)=e=(cyc(i)*(q(i)-1)+FLT(i));

```



```

kisit16(i,w)..
sum(m,x(w,i,m)*(manual(i,m)+walking(i))) =l= cyc(i);
kisit161(i,m)..                               sum(w,x(w,i,m))=e=1;
kisit17(i)..                                   cycmin(i)=l=cyc(i);
kisit24(w,i,m)..                               x(w,i,m)=l=x1(w,i);
kisit19(w,i,j)$(ord(i) ne ord(j))..           x1(w,i)+x1(w,j)=l=3-
(ka(i,j)+kb(i,j));
kisit20(i)..                                   sum(w,x1(w,i))=l=maxata(i);
model cellscheduling/all/;
Option Threads=-1;
OPTION reslim =10000000;
OPTION ITERLIM = 90000000;
Option optcr=0.02;
solve cellscheduling using mip minimizing amac ;
display cyc.l;
display x1.l;
display c.l;
display s.l;
display p.l
display b.l;
display ka.l;
display kb.l;
display x.l;
display x1.l;
display manual;
display t11.l;
display FLT;

```


ÖZGEÇMİŞ

Ad-Soyad : Ömer Faruk YILMAZ
Doğum Tarihi ve Yeri : 05.06.1989 TRABZON
E-posta : ofyilmaz@itu.edu.tr

ÖĞRENİM DURUMU:

- **Lisans** : 2010, İstanbul Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü
- **Yükseklisans** : 2012, İstanbul Teknik Üniversitesi, Endüstri Mühendisliği Ana bilim dalı

DOKTORA TEZİNDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Yılmaz, Ö. F., Durmuşoğlu, M. B.** 2017. Performance comparison and evaluation of metaheuristics for batch scheduling problem considering skilled workforce assignment in a multi-hybrid manufacturing system. *Journal of Industrial and Management Optimization*.

DİĞER YAYINLAR:

- **Yılmaz, Ö. F., Pardalos, P. M.** 2017. Minimizing average lead time for the coordinated scheduling problem in a two-stage supply chain with multiple customers and multiple manufacturers. *Computers & Industrial Engineering*, 114, 244-257.
- **Yılmaz, Ö. F., Öztayşi, B., Durmuşoğlu, M. B., Salkın, S.C.** 2017. Determination of material handling equipment for lean in-plant logistics using fuzzy analytical network process considering risk attitudes of the experts. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 24(1), 81-122.
- **Yılmaz, Ö. F., Çevikcan, E., Durmuşoğlu, M. B.** 2016. Scheduling batches in multi hybrid cell manufacturing system considering worker resources: A case study from pipeline industry. *Advances in Production Engineering & Management*, 11(3), 192.

- **Faruk Yılmaz, Ö.,** Tüfekçi, S. 2018. Handbook of Research on Applied Optimization Methodologies in Manufacturing Systems (pp. 1-449). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2944-6
- **Yılmaz, Ö. F.,** Durmuşoğlu, M.B. 2017. Evolutionary algorithms for multi-objective scheduling in hybrid manufacturing systems. *Handbook of Research on Applied Optimization Methodologies in Manufacturing Systems*. IGI Global.
- **Yılmaz, Ö. F.,** Durmuşoğlu, M.B. 2017. A methodology for order release, product scheduling and worker assignment in hybrid manufacturing systems with utility workers. *Handbook of Research on Applied Optimization Methodologies in Manufacturing Systems*. IGI Global.
- **Yılmaz, Ö. F.,** Erbiyık, H. 2016. Effective Applications of Optimization Methods in the Manufacturing Environment in Turkey. In *Comparative Economics and Regional Development in Turkey* (pp. 319-335). IGI Global.
- **Yılmaz, Ö. F.,** Baskak, M., Erbiyık, H., Budak, A. 2012. Service level within an integrated model for warehouse and inventory planning. In *Proceedings of International Conference on Industrial Engineering and Operations Management* (pp. 684-692).
- **Yılmaz, Ö. F.,** Baskak, M., Erbiyık, H. 2015. To define service level in an integrated model for warehouse and inventory planning by utilizing heuristic solution: An example. In *Industrial Engineering and Operations Management (IEOM), 2015 International Conference on* (pp. 1-8). IEEE.
- **Yılmaz, Ö. F.,** Durmuşoğlu, M.B. 2015. Hybrid of Genetic and Simulated Annealing Algorithm for Permutation Flow-Shop Scheduling Problem. *45th Annual Conference of the Italian Operations Research Society, AIRO*.
- **Yılmaz, Ö. F.,** Durmuşoğlu, M.B. 2015. Hybrid Metaheuristic Algorithm for a New Scheduling Problem in Hybrid Manufacturing Environment. ISBN:978-961-6165-45-7, SOR International Conference .
- **Yılmaz, Ö. F.,** Durmuşoğlu, M.B. 2016. A New Scheduling Problem for Hybrid Manufacturing systems in Dual Resource Constrained Environment. EURO 2016 Poznan.
- **Yılmaz, Ö. F.,** Baskak, M. 2012. Bekleyen Sipariş Durumunda Sürekli Gözden Geçirmeye Dayalı Olasılıklı (R,Q) Stok Kontrol Modeli ve Depo Yapısı.– Üretim Araştırmaları Sempozyumu (ÜAS) Gediz Üniversitesi.